Fachbereich III Informations- und Kommunikationswissenschaften

Institut für Angewandte Sprachwissenschaft

MAGISTERARBEIT

INTERNATIONALES **I**NFORMATIONSMANAGEMENT

Adapting the Multilingual Information Retrieval System MIMOR to the Characteristics of Japanese

vorgelegt von

Nina Kummer

(nkummerkasten@gmx.de)

Erstgutachter: Prof. Dr. Christa Womser-Hacker

Zweitgutachter: Dr. Thomas Mandl

Hildesheim, im Juni 2005

Abstract

This M.A. thesis describes the conception and realization of a cross-lingual information retrieval system for Japanese based on the MIMOR ("Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval") framework. After an analysis of the characteristics of Japanese and their implications for IR, an overview of established approaches and the state-of-the art in Japanese IR and cross-lingual IR with Japanese is provided. It is followed by a description of the implemented system and its integration into the existing framework. Finally, the evaluation experiments carried out with two different document genres (newspaper articles and scientific abstracts) are reported. The main focus hereby was on the testing and analysis of different indexing strategies, in particular a yomi- or pronunciation-based index in addition to conventional word-based and n-gram-based indices, and the benefits of their fusion.

Zusammenfassung

Diese Arbeit beschreibt die Implementierung eines auf dem MIMOR-Modell (Multiple Indexing for Dynamic Method-Object-Relation in Information Retrieval) basierenden IR-Systems für Japanisch. Nach einer Analyse der Besonderheiten der japanischen Sprache und ihrer Implikationen für das Information Retrieval wird ein Überblick über etablierte Strategien und Stand der Forschung im Japanisch-IR sowie im cross-lingualen IR mit Japanisch gegeben. Im Anschluss werden die implementierten Funktionen und ihre Einbindung in das bestehende System beschrieben. Die Arbeit schließt mit einer Darstellung der durchgeführten Experimente, die zur Systemevaluation mit zwei verschiedenen Dokumentgenres (Zeitungsartikel und wissenschaftliche Abstracts) durchgeführt wurden.

Der Fokus bei Implementierung und Evaluation lag auf verschiedenen Indexierungsstrategien, neben den typischerweise eingesetzten wortbasierten und n-gram-basierten Indexformen insbesondere einem aussprachebasierten Index („yomi-based index"), und ihrer Fusion.

Keywords:
Japanese information retrieval, MIMOR, cross-lingual Japanese-English IR, indexing

Table of Contents

0 Introduction

With the development of electronic data processing, the Internet, and the World Wide Web, the amount of information accessible to the modern human being has increased enormously over the last decades and will continue to do so at an even greater speed.

From the point view of information science, which defines information as *applied knowledge*, however, we are rather confronted with an exponential growth of data, which can only unfold its full potential if the right bit of knowledge is made accessible to the right person at the right time.

The information is distributed over the Web and various databases, published in many different formats, and spread out over many different languages. As the amount of data accessible through the World Wide Web increases, we need ever more sophisticated tools to organize and access this vast pool of knowledge in order to make efficient use of it. In order to turn data into information, it needs to be properly managed, and, even more important, made retrievable at will. This is achieved through information access technologies such as Information Retrieval, Information Extraction, Text Summarization, Question Answering, etc. (cf. Kando 2003).

There are currently three international conferences fostering research in these domains, namely the Text REtrieval Conferences (TREC[1]), the Cross-Language Evaluation Forum (CLEF[2]), and the NTCIR-Workshop[3]. TREC was started in 1992 in order to encourage research in information retrieval based on large test collections and to present a common basis for evaluation of different (text) retrieval methodologies.

Since fundamental IR procedures, such as tokenization or segmentation, stopping, and stemming, are language-dependent, not all the insights gained for one language or language group can be transferred to another offhand. Moreover, the distribution of information over several languages calls for a means to retrieve documents in a number of languages based on a query in a single language, that is, CLIR techniques. The first system evaluation in CLIR was held in 1997 with the Cross-Language Information Retrieval (CLIR) track at TREC.

The CLEF campaign, supported by the European Union and established in 1999, is a continuation and expansion of this track, with a focus on European languages.

Almost simultaneously, the NTCIR Workshop, its Asian counterpart, was created in Japan. It places emphasis on the enhancement of research in Information Access (IA)

[1] http://trec.nist.gov
[2] http://clef-campaign.org
[3] NII-NACSIS Test Collection for IR Systems (http://research.nii.acjp/ntcir/workshop)

technologies including but not restricted to mono- and cross-lingual information retrieval with Japanese or other Asian languages. First carried out in 1999, it is currently in its fifth round, attracting more and more international participation.

CLIR techniques are not only critical for bridging the language barrier between English and Asian languages for the sake of international information transfer, they are also frequently required for monolingual retrieval on East Asian language texts (cf. Kando 2003:5), since in some domains (e.g. scientific papers), documents containing English portions of text (e.g. abstract) may be found.

Whereas there is a close collaboration of researchers organizing the CLEF and NTCIR Workshops, with regular reports of advances on each side, there are very few systems actually being tested in both CLEF and NTCIR cross-language tasks.

The faculty of Information Science of the University of Hildesheim has been participating regularly in the CLEF workshops with its own information retrieval system MIMOR[4] since 2002. The system supports most European languages. Now, the scope of supported languages shall be extended to more “exotic” ones like Arabic and Japanese.

The goal of this M.A. thesis is the conception and realization of a cross-lingual information retrieval system for English and Japanese and its integration into the MIMOR framework.

MIMOR is modeled as an open information retrieval system designed to combine different approaches in information retrieval within one Meta system. This allows for the exploration of the performance of individual retrieval devices and/or approaches on the one hand, and also profits from the advantages of the best-performing technologies for an optimal retrieval result. These characteristics shall be utilized when tackling mono- and cross-lingual IR with Japanese-language documents.

When transferring knowledge about IR techniques from one language to another, special attention has to be paid to language-dependent IR procedures like tokenization, stopping or stemming. With Japanese lacking explicit boundaries between words in a sentence, indexing procedures are quite different from those used for European languages. Therefore, the evaluation of the performance of different segmentation techniques has always been an extensively discussed topic in Japanese IR.

Similarly to the findings in TREC, the evaluations of the NTCIR Workshop series have not produced one clearly superior system, but rather comparably well performing systems using completely different approaches. Since the first workshop in 1999, the

[4] MIMOR = Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval (cf. Womser-Hacker 1996)

question has prevailed whether complex NLP techniques or simple, language-independent n-gram indexing yield better retrieval results. These were the approaches adopted by the two top-performing systems in NTCIR-1.

The main factor influencing the performance of NLP and language-independent approaches is the percentage of words unknown to the system. N-gram approaches are very robust, whereas the performance of otherwise well-working NLP techniques drops drastically with an increasing amount of non-covered vocabulary.

These findings suggest that fusion of different approaches should be a promising strategy in Japanese IR, and that special attention should be paid to the relative performance of individual approaches with respect to certain characteristics of the document collections, such as the degree of standardization of vocabulary.

This work will therefore focus on the testing and analysis of different indexing strategies with respect to document collection properties and the benefits of merging their result lists. Further, various optimization strategies shall be tested in order to evaluate their benefits in improving the retrieval results. System training and evaluation will be carried out using parts of the NTCIR test collection.

This master's thesis presents the steps necessary for the integration of a Japanese retrieval module into the existent retrieval system and reviews the tests carried out with the system.

Chapter 1 presents the specific characteristics of the Japanese language, which need to be taken into account for Japanese IR, i.e. a complex writing system, a lack of word boundaries, the frequent use of orthographic varieties, and the difficulties that arise with respect to the encoding and processing of Japanese text.

Chapter 2 relates past research in Japanese IR and the state-of-the-art of current Japanese IR systems, focusing primarily on segmentation and indexing approaches. Furthermore, various strategies for the optimization of retrieval performance are presented.

Chapter 3 resumes approaches and resources used in Japanese/English CLIR and presents a number of translation optimization strategies.

In chapter 4, the system background is explained and the adaptations made for the integration of Japanese language support are explained.

Chapter 5 relates the experiments carried out using the NTCIR test corpus and provides an analysis of the results.

The concluding chapter sums up the findings and gives a brief outlook.

1 Linguistic and Technical Challenges of Japanese

1.1 Writing System

> *"The complex Japanese language and its writing system are inventions of the devil, designed to prevent the spread of Gospel."*
>
> Attributed to Francis Xavier (1506-1552), Spanish Jesuit missionary in Japan (Taylor et al. 1995:288)

The Japanese language is said to have the most complex writing system[5] in the world (cf. Halpern 2000). It uses a combination of four different scripts, namely the pictographic and ideographic characters adopted from Chinese (*kanji*), which have been simplified to a certain degree, two Japanese syllabaries developed out of the Chinese kanji (*hiragana* and *katakana*) with 46 basic characters each, as well as the Roman alphabet[6] (called *rōmaji*) – each script fulfilling a distinct function.

1.1.1 History

This particular situation can be better understood by tracing back the history of the Japanese writing system to its origins. Table 1 lists the significant developments in the history of the Japanese writing system.

Until the 8th century of the Japanese Nara period, Japanese did not posses a writing system of its own. Everything was transmitted exclusively orally. The Chinese characters, the *kanji*, had started arriving in Japan as early as the 1st century B.C. and were soon copied by the Japanese, but without knowledge of the system. Little by little, knowledge about the system was acquired. Chinese books, particularly Buddhist scriptures, were first brought to Japan between the 3rd and 5th centuries A.D.

5 "Writing system" refers to a type of system such as alphabet, syllabary, and logography or to a set of different scripts used to represent one language. "Script" refers to an individual form of some type of writing system, such as the Roman alphabet, the Greek alphabet, or the Cyrillic alphabet, which are all instances of the writing system "alphabet" (cf. Taylor & Taylor 1995:10).

6 "Roman alphabets" is an umbrella term for all alphabets derived from the Latin alphabet; their letters are similar, yet not identical, in shapes, names, order, number, and sound values. (cf. Taylor & Taylor 1995:114).

The first attempt of framing the Japanese language in writing, using the Chinese characters, was undertaken around the 7th century. The process of adapting the *kanji* to the Japanese language extended over several centuries.

Era	Years AD	Scripts and Literacy
Yamato	~350-710	Chinese characters and Buddhism introduced
Nara	710-794	First surviving history and poetry books in kanji
Heian	794-1185	Two forms of kana develop out of kanji; stories in kana
Muromachi	1333-1568	Romanization of Japanese by Jesuit missionary
Edo/Tokugawa	1600-1868	Some European words
Meiji	1868-1912	Many European words; new words coined on Chinese model; limit number of kanji; kana-kanji mix common; Hepburn Romanization
Showa	1926-1989	Additional kanji for education and names
Heisei	since1989	Lists of official kanji

Table 1: Significant developments in the history of the Japanese writing system (cf. Taylor et. al. 1995:281)

When borrowing Chinese characters in order to write Japanese, the Japanese also borrowed many Chinese words, to the extent that the proportion of Sino-Japanese entries in Japanese dictionaries is as high as 60 percent.

The borrowing of characters happened in three major waves, while the Chinese writing system itself was still developing (cf. 1.2 Orthographic Varieties), and involved both semantic and phonetic use of Chinese characters.

Semantic use is made when writing Japanese words with semantically equivalent Chinese characters, which correspond largely to the word unit. In this case, only the character itself is adopted and associated with the pronunciation of a genuine Japanese word.

	Chinese (ON-yomi)	Japanese (kun-yomi)
Character	車	
Reading	SHA	kuruma
Concept/Meaning	car; vehicle; wheel	

Table 2: Example of semantic use of Chinese characters for representing Japanese concepts.

The phonetic use employed only the sound of the characters, disregarding their original meaning. Those characters used as “sound” characters were simplified over time,

leading to the creation of the genuinely Japanese kana[7] syllabaries around the the 8th and 9th centuries in the following stages: Kanji → Kanji as phonetic signs → simplified Kanji shapes → Kana (cf. Taylor & Taylor 1995:308).

There are two kana syllabaries: hiragana and katakana. Katakana ("part of borrowed names") were used mainly by men in marginal notes in Chinese texts, dictionaries, and commentaries, whereas hiragana (*hira* 'without corner, popular, light') were used mainly by female authors to write letters, poems, diaries, and eventually stories. They bear a strong resemblance to the katakana characters, however, more single strokes are written in one move (cursive style), which makes them more apt for calligraphy.

The development of the kana went on until the 11th century, but the first official regularization for school purposes was not carried out until 1900.

Figures 1 and 2 show the hiragana and katakana syllabaries. The Roman transcription appears on the left, the hiragana and katakana symbols, respectively, in the middle, and the kanji from which they are believed to be derived on the right.

平仮名（ひらがな） hiragana

a	あ	安	i	い	以	u	う	宇	e	え	衣	o	お	於
ka	か	加	ki	き	幾	ku	く	久	ke	け	計	ko	こ	己
sa	さ	左	shi	し	之	su	す	寸	se	せ	世	so	そ	曽
ta	た	太	chi	ち	知	tsu	つ	川	te	て	天	to	と	止
na	な	奈	ni	に	仁	nu	ぬ	奴	ne	ね	祢	no	の	乃
ha	は	波	hi	ひ	比	fu	ふ	不	he	へ	部	ho	ほ	保
ma	ま	末	mi	み	美	mu	む	武	me	め	女	mo	も	毛
ya	や	也				yu	ゆ	由				yo	よ	与
ra	ら	良	ri	り	利	ru	る	留	re	れ	礼	ro	ろ	呂
wa	わ	和	wi	ゐ	為				we	ゑ	恵	wo	を	遠
												n	ん	无

Figure 1: The hiragana syllabary with pronunciation and original kanji[8].

[7] The etymological origin of the term "kana" is not entirely resolved. It could have developed out of karina (kari 'borrowed' and na 'name' or 'letter'), playing on the fact that the kana borrow the sounds of kanji. For more information, see [Taylor et al. 1995:307].

[8] http://www.omniglot.com/writing/japanese_hiragana.htm

Figure 2: The katakana syllabary with pronunciation and original kanji [9].

The symbols for 'wi' and 'we' were made obsolete by the Japanese Ministry of Education in 1946 as part of its language reforms. When used as grammatical particles, 'ha', 'he' and 'wo' are pronounced 'wa', 'e' and 'o' respectively. Additional sounds are represented using diacritics or combinations of syllables.

1.1.2 Usage of the Individual Scripts

Contemporary writing practice mixes Chinese characters (*kanji*) and *hiragana*, with some interspersed *katakana*. Beside these three scripts one also finds *rōmaji*, a system based on the Roman alphabet, and Arabic numbers, in Japanese texts.

Each writing system has its own function. Sometimes all four scripts can be found in one single sentence (cf. Shibatani 1992:249). The following example from a headline of the Asahi Shimbun, April 19, 2004, shows the interaction of all four scripts[10]. Kanji are marked red, hiragana blue, katakana green, rōmaji and European numerals black.

ラドクリフ、マラソン五輪代表に１万ｍ出場にも含み。

radokurifu, marason gorin daihyō ni 1 man m shutsujō ni mo fukumi
"Radcliffe, Olympic marathon contestant, to consider also appearing in the 10,000 m."

[9] http://www.omniglot.com/writing/japanese_katakana.htm
[10] http://en.wikipedia.org/wiki/Japanese_writing_system

The following paragraphs sum up the specific usage of the different scripts (cf. Hadamitzky 1995:32ff, Shibatani 1992:249, Taylor 1995:310) and point out the implications for Information Retrieval.

Kanji:

The Chinese characters make up the largest part of Japanese texts (about 50%). They represent the “concept terms” – mainly nouns and the invariable stem of adjectives and verbs – and Japanese, Korean and Chinese proper names. Since these pictographic and ideographic characters are associated with concepts or ideas, they are very efficient for concise and dense presentation of information. When Japanese native speakers read a Japanese text, they can quickly grasp its meaning by “scanning” over the kanji characters. Although the reading of a kanji may vary depending on the combinations in which it occurs, the basic meaning it represents remains basically unaltered.

From the point of view of Information Retrieval, the kanji are the most important category of Japanese characters, since they convey the meaning of words. Section 2.1, specifically sections 2.1.1 and 2.2.2, will clarify in which way kanji can be processed so as to profit from these characteristics.

Hiragana:

Hiragana are mostly used for inflectional endings of kanji concept terms (“okurigana”) and all words not written in kanji, that is, predominantly grammatical function words. In many cases, not only the actual ending, but also a part of the stem is written in Kana (cf. Okurigana Variants in 1.2.1). In some particular cases they can also represent nouns, verbs, and adjectives, e.g. when the formerly used kanji have become obsolete.

Although Hiragana make up quite a large portion of Japanese text (abound 40%) they do not play an essential role in the transmission of ideas. In general, it can be assumed that the parts of a document written in hiragana are not content-bearing. It is therefore a common practice to discard hiragana characters when indexing documents and when processing search requests.

Katakana:

In modern Japanese writing, katakana are mainly used to represent loan words and foreign proper names (except Chinese and Korean names).

They are also employed for animal and plant names (esp. in scientific language). There are some female given names (e.g. Emi, Mari) which may be written in katakana. Furthermore, the frequently employed Japanese onomatopoetica, e.g. animal sounds, children's language or exclamations are represented in katakana characters. The same holds for colloquial words or slang. Japanese telegrams use the katakana script.

The sparse usage of katakana gives them a signal effect, which is frequently exploited for typographic accentuation (e.g. in advertisements, in magazine or shop names).

Among these applications, it is especially the writing of loan words and foreign proper names that makes katakana terms most relevant for indexing purposes. Since Japanese is a modern and constantly evolving language, and moreover very prone to respond to outside influences of all kinds, there is a rapidly growing number of adopted adapted foreign words, written in katakana. English words, sounding "new, sophisticated, modish, different, or erudite" (cf. Taylor & Taylor 1995:314) to the Japanese, are now incessantly adapted to the Japanese vocabulary. There are even dictionaries devoted exclusively to them. Between 1960 and 1980, the percentage of new words adapted from foreign words increased from 43.0% to 57.6%, whereas the share of new Sino-Japanese words dropped from 40.2% to 28.8% (cf. Taylor & Taylor 1995:285). Japanese sometimes even adopts foreign expressions when there already is a genuine Japanese equivalent. With new concepts incessantly arising above all in the domains of science and technology, katakana terms are to be treated with special care when dealing with the retrieval of scientific documents.

Roman characters:

Roman characters are mainly used in tables, in writing acronyms, e.g. foreign acronyms like IMF, WWW, but also acronyms for Japanese names, such as NHK for the Japanese national broadcasting company "Nippon Hoosoo Kyookai", in proper names or for transcription or transliteration purposes, such as for indicating the names of train stations (as an aid for foreigners).

The Roman characters used are the same as those employed in Western texts, specifically the 52 upper- and lowercase letters of the Latin alphabet, sometimes decorated with accents to indicate length or tone (Lunde 1999:28). Also included are the ten numerals 0 through 9, as the Arabic numbers are frequent in horizontal writing.

Terms written in Roman characters should be kept when indexing Japanese documents, as they can represent important proper names. Special attention should be paid to the handling of acronyms.

The importance of a type of script may vary depending on the text genre. Table 3 shows an example of different distributions. A comparison between the numbers obtained from “Paper A” and “Paper B” gives an impression of the shift in the percentage of script types used from 1971 to 1982 (cf. Taylor & Taylor 1995:331). The case of the Software Manual in the right column shows that the percentage of katakana and English terms in the computer science domain is considerably higher, while the share of kanji is greatly reduced.

Script	**Paper A**	**Paper B**	**Software**
	1971	**1982**	**1991**
Hiragana	35.3%	47.1%	41.5%
Kanji	46.1	40.2	16.1
Katakana	6.1	3.9	28.8
English	0.0	0.0	6.8
Arabic Numeral	1.4	1.4	4.2
Rōmaji	0.4	0.0	0.0
Special Symbol	10.7	8.6	2.5

Table 3: Proportions of kanji, hiragana, and other characters in text (Taylor & Taylor 1995:331).

Key:
Special Symbol: items such as a short bar for a long vowel in Katakana words or the symbol for a repeated graph
Paper A: 1 million graphs in newspapers, examined from the types used for 21 days in July 1971 by Kyodo Press
Paper B: several articles from newspapers and magazines
Software: four sentences from a software manual (A&A Co. 1991)

An analysis of the NTCIR-4 and NTCIR-5 corpus (Mainichi Shimbun and Yomiuri Shimbun 1998-2001) showed the following distribution of scripts:

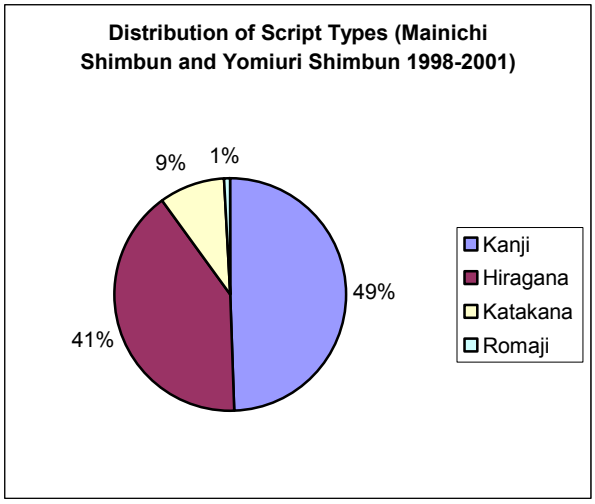


Figure 3: Distribution of script types in the Mainichi Shimbun and Yomiuri Shimbun 1998-2001.

1.1.3 The Role of Kanji for the Understanding of Japanese Text

Both kana syllabaries (*hiragana* and *katakana*) encompass the complete sound inventory of the Japanese language. Every single Japanese word or phrase can be expressed using exclusivelyut kana, but it is generally expected that the large number of homophones would lead to communication problems without the conceptual hints provided by kanji.

Owing to the very simple Japanese sound system, a great number of words share the same sound. The 5 vowels and 16 consonants add up to 21 phonemes, which is about half the number of phonemes used in English (cf. Taylor & Taylor 1995:283). This small inventory of Japanese sounds is used to produce a small inventory of extremely simple syllables. Table 4 shows the very limited patterns of possible phoneme combination in Japanese syllables (cf. Taylor & Taylor 1995:7).

Structure	English Word	Usage in Japanese
V	a	✓
CV	go	✓
VC	at	only with C = ”n”
CVC	get	only with C = ”n”
CVCC	lend	
CCVC	glad	
CCVCC	blend	
CCCVCCC	strengths	
CCVCCCC	twelfths	

Table 4: Some syllable structures (Taylor & Taylor 1995:7).

Key: V=vowel; CV=consonant-vowel; CVC=consonant-vowel-consonant

In fact, Japanese uses only about 110 syllables. English, in comparison, uses several thousands. Chinese has an inventory of about 400 syllables, which, multiplied by the four tones, yield 1,600 possible tone syllables, of which 1,300 actually occur (cf. Taylor & Taylor 1995:31). Therefore, words with the same sound abound in the Japanese language.

The native word “hana”, for example has four different meanings:

(花) flower

(端) edge

(鼻) nose

(洟) snivel

All four can be clearly differentiated by their kanji (cf. Taylor & Taylor 1995:322).

Single Kanji pronounced in monosyllabic ON (Chinese) reading tend to have many homophones, in average around 20 per syllable. In “A New Dictionary of Kanji Usage”, which defines 2,000 kanji, the syllable with the largest number of homophones, 68, is “shou”, and the next largest number, 67, is “kou”. In a larger dictionary, these numbers would be even larger. When these two Kanji are joined in “koushou”, the compound word has 20 homophones in this small dictionary, and many more in a larger dictionary. All these numerous homophones will be written in different kanji for different meanings.

Examples of homophones in the Japanese language (cf. Taylor & Taylor:323ff):

1. “shikaishikaishikaishikai” → chairmanship of the dentists’ conference
2. “kisha no kisha ga kisha de kishashita.” → Each of the four “kasha” is written in a different kanji to convey the meaning: “Your company’s reporter returned to the office by train.”

The above examples make it clear that homophones may well be the source of misunderstandings in oral communication. In fact, when listeners try to infer the meaning of an ambiguous word using speech context, they may visualize its kanji.

Critics who would like to abolish the use of kanji stress that one of Japan’s greatest literary works, the Tale of Genji (Genji Monogatari) by Lady Murasaki Shikibu in the early 11th century, was written in hiragana, with only a handful of kanji to write Chinese loan words – and understood by its readers (cf. Taylor & Taylor 1995: 307). Nevertheless, one should note kanji that convey a significant amount of semantic information which is lost in phonetic transcriptions of Japanese.

In Japanese IR, homophones nowadays do not represent a problem, as text is indexed in its written and not its phonetic representation. Before double-byte characters could be handled on a computer, however, Japanese used to be represented in a phonetic katakana transcription. In those days, IR systems also had to work with this pronunciation-based representation and were confronted with betimes ambiguous index or search terms. With the introduction of double-byte characters, this technique became obsolete. More information on pronunciation- or yomi-based indexing can be found in 2.2.5 (Pronunciation-Based Indexing).

1.1.4 Word Boundaries

Japanese text does not have spaces to mark word boundaries. The only delimiters are punctuation marks, which correspond roughly to those used in English. Furthermore, the conventionally fixed usage of kanji and kana helps in identifying the word boundaries. These cues are sufficient for the Japanese reader to identify the individual words in a text.

Word length varies considerably. According to a study by [Ogawa et al. 1995], the average character count for kanji and katakana terms is 2.51 and 8.69, respectively, varying from one to over ten characters per term. Table 5 shows the detailed statistics gained from an analysis of a total of 100,000 abstracts of Japanese Patents.

	Kanji		Katakana	
length	num. of dist. words	total frequency	num. of dist. words	total frequency
1	1120	256062	26	93
2	9983	514645	275	8889
3	23552	151983	944	83976
4	39995	148174	1401	48238
5	24717	48746	1557	23111
6	18583	30888	2526	20036
7	8533	11790	2688	12360
8	4692	6413	2030	6397
9	2075	2641	1529	4246
10	960	1124	976	2353
over 10	846	991	1729	2789

Table 5: Statistics of word lengths in Japanese patent texts (cf. Ogawa et al. 1995).

The lack of word boundaries makes it especially difficult to segment Japanese text automatically for subsequent use in the IR process. The different strategies that have been proposed and their relative performance will be presented in section 2.1.

1.2 Orthographic Variety

> *"A major factor [contributing to the complexity of the Japanese writing system] is the complex interaction of the four scripts used to write Japanese, resulting in countless words that can be written in a variety of often unpredictable ways."*
>
> (Halpern 2000)

Orthographic variety is frequent in the Japanese language and represents a serious problem for Information Retrieval, as traditional IR systems, which compare terms according to their written representation, run the risk of missing documents that contain variant forms of search terms.

This section presents the most frequent categories of orthographic variants and their origins.

1.2.1 Okurigana Variants

Okurigana are the hiragana used for grammatical endings. Unfortunately, it is not always clear how much of a verb or adjective is considered to be an ending to be written in hiragana. In many cases, not only the actual ending, but also a part of the stem is written in kana. Variants occur in the number of syllables transcribed in hiragana. On the use of okurigana, the government issues from time to time sets of guidelines defining "principal guidelines", "exceptions", "allowed uses", and "use with caution". However, those being complex, the actual okurigana use is unpredictable and depends on editorial policy or personal preference. Table 6 shows some examples and their evolution over time.

Meaning		**1959**		**1973**
to express	araWASU	表わす	arawaSU	表す
to perform	okoNAU	行なう	okonaU	行う
to catch	toraERU	捕える	toRAERU	捕らえる

Table 6: Changes in okurigana guidelines from 1959 to 1973 (cf. Taylor & Taylor 1995:311).

Table 7 presents further okurigana variations.

English	Reading	Standard	Variants
to express	kakiarawasu	書き表す	書き表わす 書表わす 書表す
handling, treatment	toriatsukai	取り扱い	取扱い 取扱
clearly	akirakani	明らかに	明に 明かに 明きらかに

Table 7: Okurigana variants (cf. Halpern 2002, Taylor & Taylor 1995:311).

1.2.2 Cross-Script Orthographic Variants

Although each of the four scripts used in Japanese has its own, well-defined function, cross-script variation is still to be found and can be quite unpredictable. The same word may be written in hiragana, katakana, rōmaji or kanji, or even a mixture of two scripts. Sometimes, an unusual script is chosen for certain words for stylistic reasons or in order to add an emotional component. Table 8 shows the most frequent cross-script variation patterns:

Type	Variants	English
Kanji vs. hiragana	大勢　おおぜい	many; crowd; great number of people
Kanji vs. katakana	硫黄　イオウ	Sulfur
Kanji vs. hiragana vs. Katakana	猫　ねこ　ネコ	Cat
Katakana vs. rōmaji	キログラム　ｋｇ	Kg
Katakana vs. hybrid	ワイシャツ　Ｙシャツ	shirt (trans: white shirt); business shirt
Kanji vs. katakana vs. hybrid	皮膚　ヒフ　皮フ	Skin
Kanji vs. Hybrid	彗星　すい星	Comet
Hiragana vs. katakana	ぴかぴか　ピカピカ	glitter; sparkle

Table 8: Cross-script variants (cf. Halpern 2002, Halpern 2003).

1.2.3 Katakana Variants

In recent years, there has been a sharp increase in the use of katakana, the script employed for writing loanwords, especially in technical terminology. As a modern and living language, Japanese is constantly evolving, producing new expressions for new concepts. These are very often the adaptations of foreign, mostly English terms. Unfortunately, katakana orthography is often irregular. Since the katakana syllabary is used to transcribe the phonetic structure of foreign words, the orthography often depends on the interpretation of the correct pronunciation, which may vary from person to person. Figure 4 shows the complete inventory of katakana syllables.

ga	ガ	gi	ギ	gu	グ	ge	ゲ	go	ゴ	kya	キャ	kyu	キュ	kyo	キョ		
za	ザ	ji	ジ	zu	ズ	ze	ゼ	zo	ゾ	gya	ギャ	gyu	ギュ	gyo	ギョ		
da	ダ	ji	ヂ	zu	ヅ	de	デ	do	ド	nya	ニャ	nyu	ニュ	nyo	ニョ		
ba	バ	bi	ビ	bu	ブ	be	ベ	bo	ボ	hya	ヒャ	hyu	ヒュ	hyo	ヒョ		
pa	パ	pi	ピ	pu	プ	pe	ペ	po	ポ	bya	ビャ	byu	ビュ	byo	ビョ		
sha	シャ			shu	シュ	she	シェ	sho	ショ	pya	ピャ	pyu	ピュ	pyo	ピョ		
ja	ジャ			ju	ジュ	je	ジェ	jo	ジョ	mya	ミャ	myu	ミュ	myo	ミョ		
cha	チャ			chu	チュ	che	チェ	cho	チョ	rya	リャ	ryu	リュ	ryo	リョ		
fa	ファ	fi	フィ	fu	フ	fe	フェ	fo	フォ								
va	ヴァ	vi	ヴィ	vu	ヴ	ve	ヴェ	vo	ヴォ								

Figure 4: Complete inventory of katakana sounds[11].

The katakana for the initial "v" are recent creations. This sound used to be represented with the syllables with an initial "b", and some people still prefer to use those katakana.

Table 9 shows the major types of katakana variation.

[11] http://www.omniglot.com/writing/japanese_katakana.htm

Type	English	Reading	Standard	Variants
Macron[12]	computer	konpyuuta konpyuutaa	コンピュータ	コンピューター
	user	yuuza yuuzaa	ユーザー	ユーザ
Nakaguro[13]	online	onrain	オンライン	オン・ライン
	ice cube	aisukyuubu	アイスキューブ	アイス・キューブ
Long vowels	eye shadow	aishadoo	アイシャドー	アイシャドウ
	maid	meedo	メード	メイド
Multiple kana	Diesel	diizeru jiizeru	ディゼル	ジーゼル ヂーゼル
	team	chiimu tiimu	チーム	テイーム
	violin	baiorin vaiorin	バイオリン	ヴァイオリン
Small katakana variants	quota	kuootaa kwootaa	クオーター	クォーター
Others	Jerusalem	ierusaremu	エルサレム	イェルサレム

Table 9: Katakana variants.

1.2.4 Hiragana Variants

Although hiragana orthography is generally regular, there are some irregularities, mainly due to evolution in the orthographic rules over time. Table 9 shows the two most frequent variations in hiragana orthography.

Type	English	Reading	Standard	Variants
Traditional	big	ookii	おおきい	おうきい
づ vs. ず	continue	tsuzuku	つづく	つずく

Table 10: Examples of hiragana variants.

[12] dash-like symbol indicating long vowels
[13] Middle dot between katakana words

1.2.5 Kanji Variants

Although the Japanese writing system underwent major reforms in 1946 and 1981, and the character forms have now been standardized, there is still a significant number of variants in common use. Frequently used and complex Chinese characters have been simplified over time – sometimes in several ways. Their traditional forms also continue to exist, especially in proper nouns and classical works.

1.2.6 Phonetic Substitutes

There is a large number of orthographic variants in Japanese that are based on the principle of "phonetic substitution". There are cases in which two characters are interchangeable in certain compounds. The original character and the phonetic replacement character share the same reading and are often similar in meaning. Although the use of the obsolete characters is gradually ebbing away, they are still employed rather frequently.

English	Reading	Phonetic Replacement	Phonetically Replaced
fermentation	hakkoo	発酵	醱酵
satire	fuushi	風刺	諷刺
linking	renkei	連係	連繋
linking	moosoo	盲想	妄想
abuse	ranyoo	乱用	濫用

Table 11: Phonetic substitutes (cf. Halpern 2003).

1.2.7 Orthographic Variety and IR

One possible solution for the automatic handling of orthographic variety, suggested by [Halpern 2000, 2003], is lexicon-based disambiguation – a comprehensive dictionary of all variant forms along with an algorithm that performs a simple table-lookup and normalization of all variant forms to a base form. However, such a dictionary is costly to compile, and requires constant maintenance, as the language is evolving quickly. Therefore, a more flexible strategy for automatic handling of orthographic varieties is needed.

From the information retrieval point of view, we can classify orthographic varieties in Japanese into two groups:

1. Variants originating from a different written representation of the same phoneme (cross-script variants, okurigana variants, hiragana variants, kanji variants, and phonetic substitutes).

2. Variants originating from a different interpretation of the sound structure to be represented (katakana variants).

Variants in the first group share the same pronunciation. This fact can be exploited for information retrieval, if the terms are matched using their pronunciation instead of their written representation. Variants of the second group need a different treatment. When foreign words are transcribed into katakana, the nearest Japanese sound substitutes for any sound not available in Japanese. There is only a limited number of ambiguous cases, where there is more than one transcription of a foreign sound (e.g., キ /ki/ and ク /ku/ for the rendering of the English sound /ik/ as in “cake”). Consequently, katakana variants still share most syllables, and only differ in minor aspects (i.e., one or two characters). Due to this phenomenon,matching terms based on their editing distance might be an effective means of retrieving documents containing katakana variants of a search term.

1.3 Character Encoding Issues

The enormous number of (kanji) characters used for writing Japanese leads from a linguistic to a technical challenge for information processing: 7- or 8-bit code space with 128 and 256 code points respectively, as used for Western languages, is not sufficient to cover the tens of thousands of Japanese characters[14]. The following section shall describe the problems that arise from having such a large number of characters to deal with, the solutions found and the consequences for the design of multilingual applications.

1.3.1 Character Sets

Character sets are a prerequisite for information processing and character encoding. A character set can be thought of as a “common bucket of characters” (cf. Lunde:6f)

[14] The actual number of existing kanji is somewhere between 40,000 and 60,000. The current advisory Japanese character set, called Jōyō (“daily use”) Kanji, contains 1,945 characters (Lunde 1999:7). There are another 285 kanji used for writing personal names.

whose main function is to limit the number of characters which need to be learned. This is not a real issue in languages like English, where there is only a very restricted number of characters. However, with languages like Japanese, with tens of thousands of characters in use, sometimes even in various, simplified or historical versions, the definition of a common-use character set becomes vital. In 1949, the Japanese government published the current character set, called *Jōyō Kanji* ("daily use kanji"), containing 1,945 characters. There are another 285 kanji sanctioned for use in writing personal names, called *Jinmei-yō Kanji* ("personal name use kanji") (cf. Lunde 1999:69f). These character sets were defined with education in mind and are referred to as non-coded character sets.

1.3.2 Coded Character Sets

For text processing on computer systems, coded character sets are necessary in order to guarantee the correct display of documents. ASCII is a Western character set standard which ensures the communication between (Western) computers. While one byte is sufficient for the representation of the limited number of Western characters, coded character sets for Asian languages need to break this one-byte limit and resort to multiple byte representations.

The first multiple-byte coded character set standard for Japanese, called JIS C 6626-1978, was established by the Japanese Standards Association (JSA) in 1978. The JIS C 6626-1978 has been modified a number of times over the years. The current version is JIS X 0208:1997. It is considered the most basic Japanese coded character set and is widely implemented on a variety of platforms (cf. Lunde 1999:106). Apart from this standard national character set, there also exist a number of deviating character set standards, such as international or proprietary vendor character set standards.

When designing multilingual applications, it is vital that the individual character sets of all concerned languages be covered. The ASCII (American Standard Code of Information Interchange) can be regarded as such a standard for writing English and most European languages that share the same set of characters. Unfortunately, there is no comparable, universally recognized or accepted character set standard for Asian languages. However, Unicode can be considered a good first attempt (cf. Lunde 1999:2).

Unicode has been developed in order to provide a single repertoire of characters for most of the world's written languages[15].

[15] Unicode is a subset of the ISO 10646-1:1993 standard (cf. Lunde 1999:121).

1.3.3 Character Encodings

The mapping of a numeric value to a character, the character encoding, is the next step for making text interpretable by computers. Again, there is no universally recognized encoding method for Asian language characters. And again, the various Unicode encodings can be considered an attempt at accomplishing this. The most commonly used encodings for Japanese are ISO-2022-JP, EUC-JP and Shift-JIS.

The ISO-2022 encoding method, as a modal encoding method[16] not being very efficient for internal storage or processing on computer systems, is used primarily as an information interchange code for moving text between computer systems, e.g. via email.

EUC (Extended Unix Code) encoding is implemented as the internal code for most Unix software configured to support Japanese. It was developed for handling multiple character sets, Japanese and otherwise, within a single text stream.

The Japanese instance of EUC is called EUC-JP and was standardized in 1990. According to [Lunde 1990:159], the trend in software development is to produce systems that process EUC-JP, because it is much more extensible than Shift-JIS. Naturally, most Unix operating systems process EUC-JP encoding internally.

Shift-JIS encoding, originally developed by Microsoft, is widely implemented as the internal code for a variety of platforms, including Japanese PCs and MacOS-J. It has now become part of the JIS X 0280 standard. JIS X 0208:1997 contains its definition.

Both EUC and Shift-JIS are non-modal encoding methods[17].

International encoding methods employed for encoding Unicode are UCS-2, UTF-7, UTF-8, and UTF-16. A detailed description of these can be found in [Lunde 1999:186ff]. The most pure representation of Unicode Version 2.0 or higher is UTF-16.

The design of a multilingual information retrieval system with support for the Japanese language therefore faces a twofold challenge: firstly, since there is no universally accepted character encoding scheme for Japanese, it might be necessary to check and possibly convert the encoding of the documents before further processing them automatically. Secondly, the system's internal representation should use Unicode and

[16] Modal encoding methods, such as ISO-2022, use escape sequences or special characters in order to switch between character sets or different versions of the same character set. Encoding takes place in two stages – first, mode-switching (initiated by one of the aforementioned markers), second, the handling of the actual bytes representing the characters. Modal encoding methods generally use 7-bit bytes.

[17] Non-modal methods make use of the numeric values of bytes in order to decide when to switch between one- and two-byte modes. These methods usually use eight-bit bytes, enabling the eight bit of a byte, and are typically variable-length. The most common examples of non-modal encoding methods are EUC and Shift-JIS.

one of its encoding methods in order to cover the character sets of all supported languages.

2 Past Research in Japanese Information Retrieval

2.1 Segmentation Strategies

As explained in section 1.1.4, Japanese text is written as a linear sequence of characters without spaces or other markers to indicate word boundaries. Therefore, unlike in English IR, words cannot directly be used to index text[18]. As a consequence, in Japanese IR, the segmentation of text has to be tackled first in the indexing process. The search for accurate methods to segment Japanese text has thus always been a main focus in Japanese IR. [Kando et al. 1999] even regret in the proceedings of the first NTCIR workshop that *"[T]raditionally, [the] Japanese IR community have tended to pay too much attention to the methods of segmenting texts into tokens rather than retrieval models or algorithms themselves."*

Text segmentation methods can be classified into two categories according to the unit of segmentation: n-gram or character-based and word-based approaches[19]. The word-based approaches can be further divided into morphological, dictionary-based, and statistical segmentation.

2.1.1 Morphological Analysis

The most language-specific segmentation approach is segmentation using a morphological analysis tool. A morphological analyzer has to incorporate detailed and complex linguistic knowledge and is generally heavy and complex software. Integrating a morphological analyzing tool complicates the IR system. Moreover, morphological analysis takes quite some time and slows down the indexing process considerably (cf. Ogawa & Yasushi 1997).

[18] This problem of missing word boundaries exists in a less obvious form in non-Asian languages with compound words.

[19] [Fuji & Croft 1993] mention a third basic indexing technique besides word-based and n-gram-based approaches, called subcharacter-based indexing. This approach is based on the fact that the majority of the kanji characters are not the minimum semantic units in a word. They are composed of parts which are called bushu. In subcharacter-based indexing these are extracted as indexing terms. Also problematic is the superfluous nature of some bushu and the rare occurrence of others, which leaves the semantics too vague. Therefore, this approach has not been further pursued.

The most widely used analysis tool among participants of the NTCIR-4 workshop was ChaSen[20] (cf. Kishida et al. 2004). It evolved from an earlier tool called JUMAN and has been continuously improved over the years. The current version is chasen-2.3.3. ChaSen segments Japanese text into morphemes, tagging these with their parts-of-speech (POS) and pronunciation. [Matsumoto et al. 2000] give a brief description of the functioning and parameters.

The POS information provided in the output of a morphological analyzer can be used to eliminate word classes with little semantic value, such as conjunctions or particles.

A serious disadvantage of segmentation by morphological analyzers is that these tools use a pre-defined list of words for morpheme identification and may completely fail to capture the words that are not contained in this list (cf. Takeda et al. 2002). This is commonly referred to as the "Out-of-vocabulary problem". Another problem arises when word boundaries are ambiguous. This is especially the case with strings containing compound words or phrasal terms, as the morphological analyzer will tend to extract the individual component words as separate units, if there are entries for the individual component words in the dictionary. [Yoshioka et al. 2001] have analyzed the results of Japanese morphological analysis and pointed out several inconsistencies and the consequences for information retrieval. They further suggest that a morphological analyzer for an IR system should be able to segment text on two levels – on the word-level as well as on the level of phrase identification.

2.1.2 Dictionary-Based Segmentation

Since morphological segmentation implies the use of a dictionary, the term dictionary-based segmentation is sometimes used as an umbrella term for morphological analysis and literal dictionary-based methods (e.g. Jones et al.1998). In the proper dictionary-based approaches, a kind of table-lookup is carried out in order to match the input string to the contents of a dictionary. Dictionary-based algorithms can be classified into three groups: longest match, shortest match and overlap match (cf. Huang et al. 1998).

In the longest match approach, the longest matching strings are extracted and shorter tokens within are discarded. Since longer tokens in the dictionary are more specific, longest match segmentation produces fewer tokens with more specific meaning. This method runs the risk of losing short, but valuable index terms.

Shortest match algorithms extract the first matching tokens and accordingly create more tokens with less specific meaning, which may subsequently hurt precision.

[20] http://chasen.aist-nara.ac.jp/hiki/ChaSen/

The overlap match approach tries to overcome the disadvantages of the former methods by allowing overlapping tokens, so that both the compound and its constituent words can be extracted. Unmatched strings without equivalent in the dictionary can be used as single-character tokens for indexing and searching. They can also be used to expand the dictionary (cf. Huang et al. 1998).

The dictionary-based approach shares with the morphological analysis approach a heavy dependence of its accuracy on the coverage of the dictionary. This is a serious drawback, since it is unfeasible to build an exhaustive dictionary. [Ogawa & Matsuda 1997] speak of a "large dictionary that ideally includes all the words in the universe". It is especially difficult to list such types of words as proper nouns, acronyms and technical terms – all very important for IR. A good dictionary requires constant maintenance, which is an expensive task and often lags behind the appearance of new words (cf. Jones et al. 1998).

2.1.3 Statistical Segmentation

Statistical segmentation, the third basic word-based approach, also uses words as indexing units, but identifies them using statistical information instead of a dictionary and linguistic knowledge (cf. Ogawa & Matsuda 1997). The statistical data is collected by using a morphologically analyzed corpus in which words are identified manually or automatically, such as the POS-tagged corpus provided in the Second NTCIR Workshop. Knowledge about the four different character classes used in Japanese writing and their grammatical functions and usage patterns can be helpful when defining segmentation likelihood. A switch between character classes indicates in most cases a break between two words. The main exceptions are conjugational parts; that is kanji with inflectional hiragana. Those can be neglected since they do not play an important role in IR.

The statistical segmentation method does not require a dictionary and is robust against unregistered words. Expensive, constant maintenance is therefore not required. However, just as dictionary-based segmentation depends on the coverage of the dictionary, statistical segmentation depends heavily on the quality and coverage of the training data (cf. Nie et al. 1996). Statistical segmentation is usually less accurate than dictionary-based segmentation, which results in less effective retrieval (cf. Ogawa & Matsuda 1997).

[Takeda et al. 2002] successfully applied a statistical segmentation method based on the adaptation measure[21]. This approach does not require a segmented training

[21] "Adaptation" is a statistical measure based on the Adaptive Language Model. It can be described as the conditional probability of word w to appear repeatedly when it is contained in a certain document. The

corpus. In experiments with the NTCIR-1 and 2 collections, the statistical segmentation method led to a better retrieval performance than both manual and morphological segmentation (ChaSen)[22].

The statistical segmentation approach is not investigated by many research groups in Japan (cf. Ogawa & Matsuda 1997).

A problem that all word-based approaches are confronted with is very well described by [Jones et al. 1998]:

> *"Ideally, we would like to automatically perform a perfect segmentation of the text into its constituent words. Once the words were available, existing retrieval techniques could easily be explored. However, such perfect segmentation is not possible; indeed it is sometimes not even clear what the definition of individual words should be."*

As the main source for this segmentation ambiguity they cite the free generation of new compound nouns. In Japanese, new words are easily generated by compounding existing words. This is especially frequent for the designation of complicated meanings and objects. As a result, the same idea may be expressed using a number of different words. There often is no one ideal segmentation solution for those compounds. A possible solution may be to use both the specific compounds ("long-unit-keyword") and their constituents ("short-unit-keyword"), as the compound expressions often share component words (cf. Ogawa et al. 1993).

2.1.4 N-gram Segmentation

N-gram segmentation methods avoid this ambiguity altogether and completely give up the notion of obtaining words as index terms. Indeed, correctly segmented words are not necessarily the most effective indexing unit. As [Huang & Robertson 1997] observe, the goal of segmentation for IR purposes is "obtaining good indexing features rather than correct segmentation".

N-gram indexing ignores word boundaries and extracts overlapping sequences of n successive characters from the input string as the indexing units. Retrieved docs are ranked according to weights of n-grams instead of words. The n-gram approach is

value of adaptation is different with content and function words and contains information on boundaries of chunks of words. The exact formula and further information can be found in [Takeda et al. 2002].

[22] ChaSen showed the poorest performance, which may be explained by the fact that the test collection contains rather specialized technical documents and the vocabulary was not sufficiently covered by the dictionary.

completely language-independent and is also being investigated in non-Asian language IR[23].

In Japanese IR, it has commonly been found that n=2 yields the best retrieval results. This is due to the fact that words consisting of two (kanji) characters are the most frequent (cf. Figure 3 in section 1.1.2).

Character-based or unigram segmentation can be regarded as a special form of n-gram segmentation where n=1. Due to the poor semantic value of hiragana characters, they are often discarded before n-gram segmentation (cf. Chen et al. 1999, Savoy 2004). [Chen & Gey 2003] also discard single-byte Roman characters, creating n-grams only from fragments of either kanji or katakana. [Fuji & Croft 1993] experimented with unigram segmentation of kanji strings, dropping all hiragana from the text and keeping sequences of katakana characters or English characters as index terms on a term basis.

The advantage of the n-gram approach is that, as a completely data-driven technique, it is free from the problems of morphological and dictionary-based segmentation methods (cf. Chow et al. 2000). It does not require any maintenance and there is no risk of losing words by parsing errors. It is also very easily implemented and has largely proven to be at least as effective as dictionary-based word indexing (cf. Fuji & Croft 1993, Chen et al. 1999)[24]. Due to its absolute independence of language-specifics, the n-gram approach may also be an option for multilingual IR systems. [Juang & Tseng 2002], for example, were able to prove the robustness of the n-gram approach by applying a system which was originally designed for Chinese to Japanese and Korean IR.

However, the biggest advantage of n-gram segmentation is at the same time its biggest drawback: Because words are not recognized on principle, word-level semantics are completely missed. Consequently, a pure n-gram approach is not very suitable for techniques like Query Expansion, where the query is to be expanded with terms related to the search terms, or Pseudp Relevance Feedback (PRF), where significant terms are extracted from search results (cf. Sato et al. 1999). PRF will be explained in section 2.3.2.

[23] In some respect, the segmentation problem of Asian languages can be compared to the decomposition of compounds in some non-Asian languages such as German. In languages where the use of compounds is frequent and the correct or best decomposition not always evident, n-gram approaches may be a solution. However, in languages with obvious word boundaries, n-grams are normally only created within a word, not crossing word-boundaries.

[24] Comparing overlapping bi-gram segmentation of kanji and katakana text fragments and dictionary based segmentation, [Chen et al. 1999] find that the simpler bi-gram segmentation method outperforms the dictionary based segmentation by more than 30%. However, they emphasize that this result is due to both the incompleteness of the dictionary and its phrasal nature.

Equally, basic n-gram segmentation is not suited for cross-language retrieval, where a translation step needs to be carried out on the word level.

A disadvantage of n-gram as an index unit is the high storage cost of bi-grams. An n-gram based index is considerably larger than a word-based index, especially in Japanese with about 2,000-3,000 different characters in common use (cf. Fuji & Croft 1993). This issue will be discussed in greater detail in the next section.

As has been shown, both word-based and character-based segmentation have their strengths and drawbacks. In the next section, the effectiveness of different indexing methods will be analyzed. As segmentation strategy often dictates index type, segmentation and indexing approaches are often not completely independent from each other.

2.2 Comparison of Indexing Strategies

The indexing strategy is the process responsible for the extraction or generation of index units. Since documents are generally selected based on a similarity measure computed from the weights of matching indexing units, the choice of indexing strategy and indexing unit has a great effect on retrieval effectiveness (cf. Ogawa & Matsuda 1997).

2.2.1 Specificity and Exhaustivity

There are two factors affecting retrieval performance, namely exhaustivity (the degree to which the index covers the topics of the documents) and specificity (the precision of the index), which are directly linked to recall and precision, respectively.

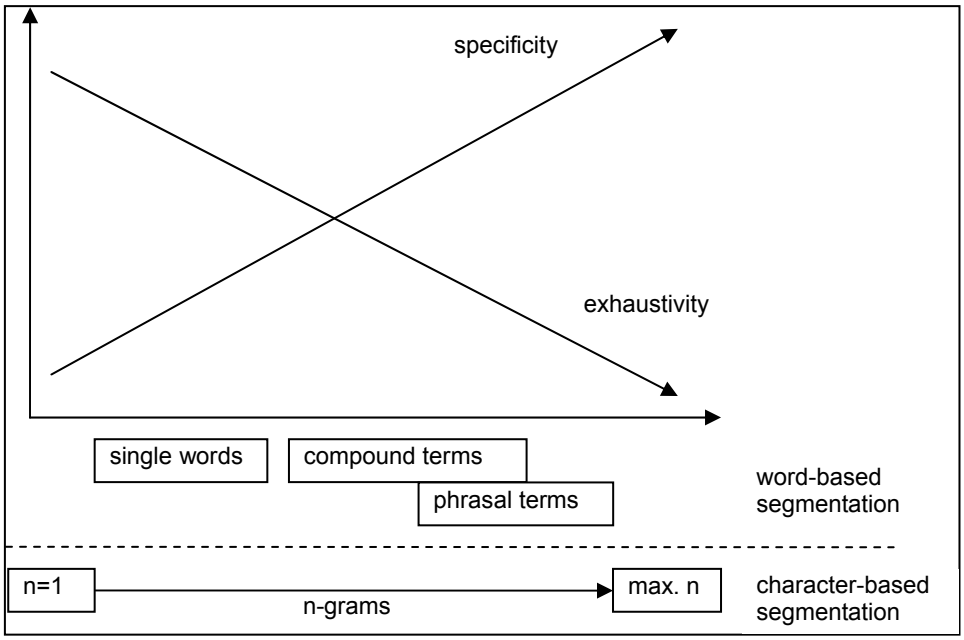


Figure 5: Trade-off between specificity and exhaustivity according to types of indexing units.

As can be seen in Figure 5, specificity rises with the length or complexity of the indexing unit. Single terms have less discriminant power than compound terms, and phrasal terms are generally even more specific than compound terms.

[Fujita 1999] points out that “a word is often too vague and ambiguous to be used as [an] indexing unit in isolation, in other words, single words are not a very meaningful unit in specific domain terminology, but no more than a part of larger units i.e. compound words, which are more meaningful information carriers in the domain terminology.”

Also noted is that not taking local syntactic relations in noun phrases into consideration might lead to a loss of information. Using only the single word term “情報” (information) and “管理” (management), for example, the system could not distinguish “情報管理” (information management) from “管理情報” (information for management). However, while its discriminant power rises, the indexing unit may become too specific to reflect relationships between terms. Consequently, exhaustivity suffers.

Equally, the specificity of n-grams rises with the length of the n-gram, with a concurrent decrease of exhaustivity. If n is too large and smaller n-grams contained within cannot be accessed, this will gravely affect recall.

In topic 50 of the NTCIR-4 workshop, for example, there is a query term “地下核実験” (underground nuclear testing), which by a word-based segmenter is split into “地下” (underground), “核” (nuclear), and “実験” (experiments). The n-gram segmenter, however, only creates segments of two characters and therefore would not match all

occurrences of the 1-character word “核” (nuclear). This prevents it from matching related text such as “核拡散防止条約” (Nuclear Non-Proliferation Treaty), “核兵器”(nuclear weapons) or “核開発” (nuclear development) (cf. Tomlinson 2004).

Due to the ideographic nature of kanji characters, unigram indexing can produce so-called thesaurus effects: two words that share the same kanji character often have some conceptual relationship (cf. Fuji & Croft 1993). This can be exploited in order to control the problem of synonyms.

Figure 6: Examples of thesaurus effects (cf. Fuji & Croft 1993).

The danger of small n-grams are false drops. False drops occur, when the characters do not appear sufficiently close to each other in the document, i.e. in the same word or phrase. This results in poor precision and may confound users. To overcome the lack of specificity, [Fuji & Croft 1993] suggest post-coordination (i.e., run-time) handling.

2.2.2 Computational Cost

Apart from the retrieval effectiveness, two more factors may play a role in the evaluation of an indexing strategy: the storage requirement of the resulting index and the time needed for computation. Their importance depends on the context of the individual system.

A serious drawback of n-gram approaches is the large overhead they produce and the resulting high storage cost of the index. This is especially problematic for languages with a large character set, such as Japanese. With about 7,000 different characters in

Japanese (cf. Ogawa & Matsuda 1997), overlapping bi-gram segmentation might extract up to 49 million bi-gram terms (cf. Lin et al. 2000)[25].

Regarding computational time, character-based indexing is considerably faster than word-based indexing. Depending on the size of the n-gram and the index structure, however, n-gram retrieval may be slower than word-based retrieval.

[Ogawa & Matsuda 1997] note that bi-gram indexing cannot efficiently handle single-character words, which are frequently found in Japanese (cf. Ogawa & Iwasaki 1995). To process a single character query like "木" (tree), for example, all the bi-grams that contain "木", such as "大木" (big tree), "樹木" (which also means "tree") and so on, must be retrieved.

With about 7,000 different characters in Japanese[26], there are potentially 14,000 bi-grams containing "木". Even if not all of the 14,000 bi-grams appear in real text, there is still a large number of bi-grams that need to be processed for a single character query, resulting in slower retrieval.

2.2.3 Case-by-Case Analyses

With regard to the definitive superiority of word-based or character-based segmentation and indexing, no consensus has yet been reached, as studies have repeatedly reported very similar performance of very different approaches. Comparisons between character- and word-based indexing approaches are complicated due to the difficulties of segmenting Japanese text (cf. Yoshioka et al. 2001).

The tenor seems to be that in spite of its relative simplicity, from the point of view of information retrieval performance, n-gram-based indexing is at least as effective as word-based indexing:

[Fuji & Croft 1993]: "[…] the character-based indexing performed retrieval as well as, or slightly better, than the word-based system[27]."

[Chen et al. 2001]: "The experimental results show the bi-gram indexing outperformed the word-based indexing […]."

[Chen & Gey 2002]: "The performance of word indexing and that of unigram-and-bi-gram indexing suggest that both indexing methods are equally effective[28]."

[25] [Fuji & Croft 1993] speak of 2,000 to 3,000 characters in practical use, which would result in 4 to 9 million potential bi-gram terms, still a very high number.

[26] See [27]

[27] Words were decompounded by the ChaSen predecessor JUMAN.

[Tomlinson 2004]: “After decompounding, the differences between segmenting into words and an overlapping n-gram approach were not statistically significant […].”

Whereas most groups only compare the average performance, [Ozawa et al. 1999] carried out a more detailed analysis of the performance of the respective approaches with individual topics and concluded that the advantage of each method depends on the specific circumstances of the task.

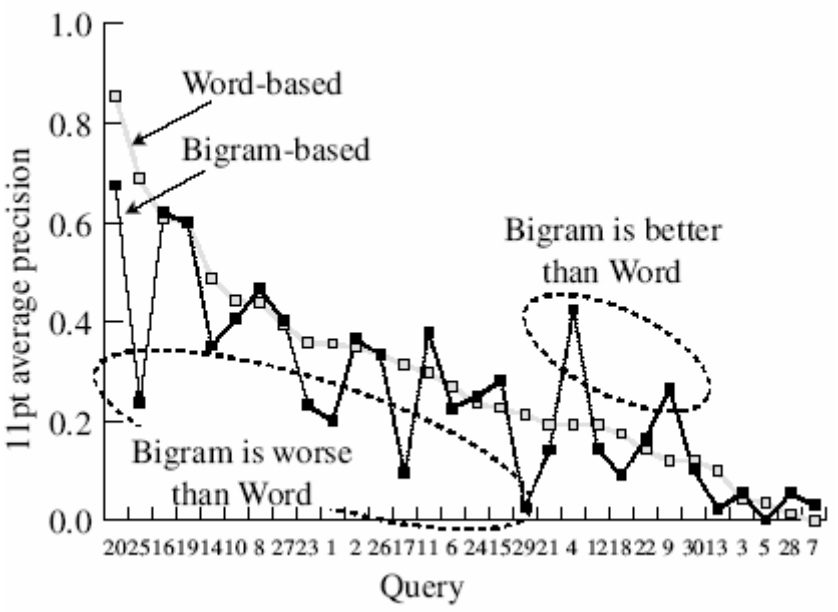


Figure 7: Performance of word- and bi-gram-based indexing per topic.

A subsequent analysis of the topics with the largest difference in performance yielded some interesting results:

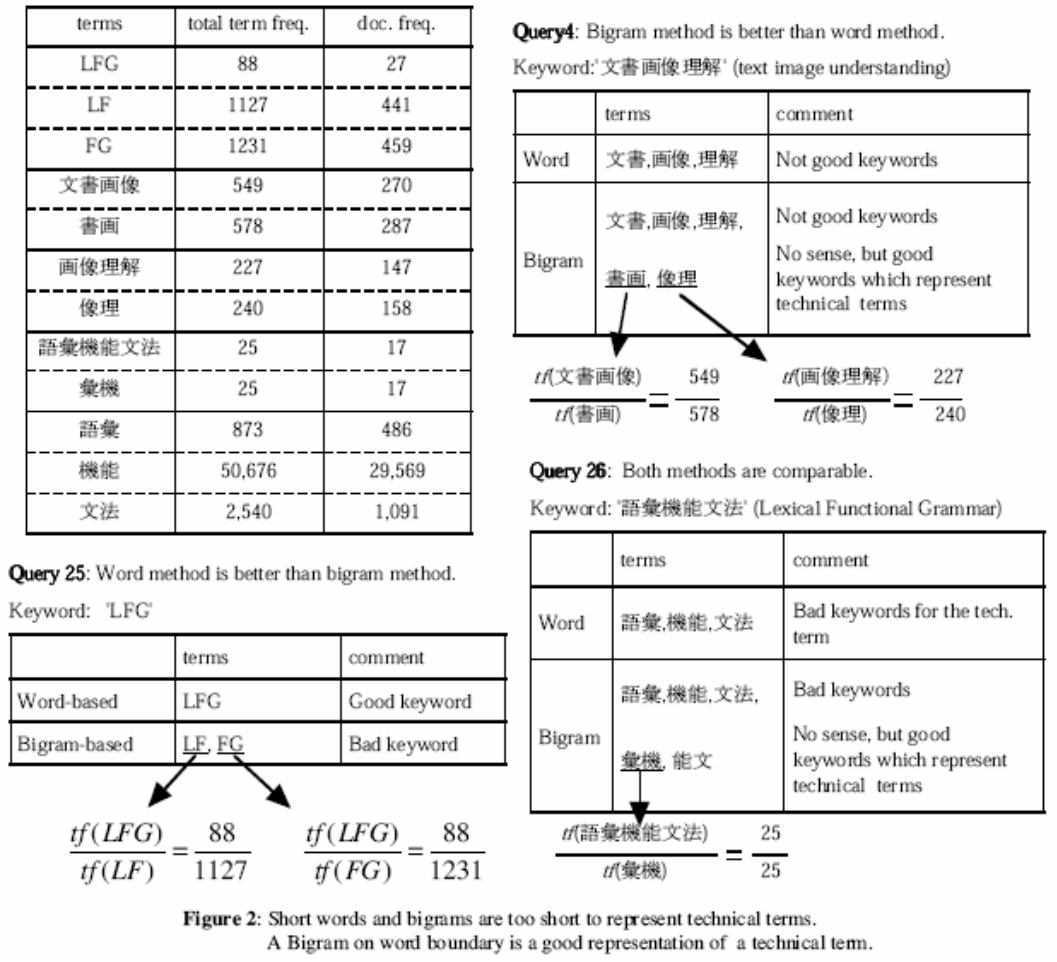
In topic 25, for example, bi-gram indexing performed very poorly, whereas the word-based method performed satisfactorily. The reason was found to be the different handling of the term “LFG” (the abbreviation of the linguistic term “Lexical Functional Grammar”). The term “LFG” is a good keyword, but the bi-gram method uses only parts of it (‘LF’, ‘FG’). Those constituent bi-grams, however, occur too frequently in the text and do not have a sufficient discriminant value.

The opposite was the case in topics 4 and 9. Here, the bi-gram approach performs considerably better than the word-based approach. This is due to the different

[28] The word index was created using the morphological analyzer ChaSen. For the n-gram index, the texts were split into single-character unigrams and overlapping bi-grams consisting of only kanji and katakana characters.

segmentation of the key-phrase: “文書画像理解” (text image understanding). The word-based segmentation splits this compound into “文書” (text), “画像” (image), and “理解” (understanding).

In this case, bi-gram segmentation works better than three words, because the individual words are too general. The (meaningless) bi-gram “書画”, however, refers in 90% of the cases to “文書画像” and is therefore highly selective. Figure X sums up the mentioned examples.



terms	total term freq.	doc. freq.
LFG	88	27
LF	1127	441
FG	1231	459
文書画像	549	270
書画	578	287
画像理解	227	147
像理	240	158
語彙機能文法	25	17
彙機	25	17
語彙	873	486
機能	50,676	29,569
文法	2,540	1,091

Query 25: Word method is better than bigram method.

Keyword: 'LFG'

	terms	comment
Word-based	LFG	Good keyword
Bigram-based	LF, FG	Bad keyword

$$\frac{tf(LFG)}{tf(LF)} = \frac{88}{1127} \qquad \frac{tf(LFG)}{tf(FG)} = \frac{88}{1231}$$

Query4: Bigram method is better than word method.

Keyword:'文書画像理解' (text image understanding)

	terms	comment
Word	文書,画像,理解	Not good keywords
Bigram	文書,画像,理解, 書画, 像理	Not good keywords No sense, but good keywords which represent technical terms

$$\frac{tf(\text{文書画像})}{tf(\text{書画})} = \frac{549}{578} \qquad \frac{tf(\text{画像理解})}{tf(\text{像理})} = \frac{227}{240}$$

Query 26: Both methods are comparable.

Keyword: '語彙機能文法' (Lexical Functional Grammar)

	terms	comment
Word	語彙,機能,文法	Bad keywords for the tech. term
Bigram	語彙,機能,文法, 彙機, 能文	Bad keywords No sense, but good keywords which represent technical terms

$$\frac{tf(\text{語彙機能文法})}{tf(\text{彙機})} = \frac{25}{25}$$

Figure 2: Short words and bigrams are too short to represent technical terms.
A Bigram on word boundary is a good representation of a technical term.

Figure 8: Short words vs. bi-grams as index terms.

Other detailed case-by-case observations, based on the NTCIR-4 corpus, can be found in [Tomlinson 2004].

2.2.4 Enhanced Indexing Approaches

In order to take maximal advantage of the strengths of the individual approaches, while at the same time minimizing their disadvantages, a number of enhanced approaches have been suggested. These range from methods to reduce the index size of character-based systems to methods tackling the exhaustivity-specificity tradeoff to hybrid approaches. The following paragraphs will contain a description of the most interesting of these enhanced approaches.

Enhanced n-gram approaches:

[Ozawa et al. 1999], based on the hypothesis that simple bi-grams are insufficient in technical language, where word length increases, proposed an adaptive n-gram segmentation method that removes noisy n-grams and changes the length of n-grams according to the similarity between the query and the document[29]. This method helps to reduce the size of the index while focusing only on effective keywords. It functioned especially well for the queries containing long technical terms, such as "データマイニング" ("data mining" in katakana).

In comparison, the adaptive (document-specific) methods worked better than non-adaptive methods. The adaptive n-gram approach also showed better performance than a word-based approach.

[Chen et al. 2001] tested an alternative segmentation method that breaks text into one-character terms and two-character terms that do not overlap with each other, in order to avoid large index files caused by overlapping bi-gram indexing. The segmentation with the highest probability is chosen based on word occurrences in the collection.

[Sato et al. 2001] experimented with long-gram-based indices stored in a tree structure that allows the retrieval of index strings shorter than the gram and reduces the rate of false drops. By further coding the gram in a compact way, they could reduce the index size considerably.

[29] The best segmentation is computed with the Viterbi algorithm, which uses a lattice to select the n-grams with the maximum term frequency within the document. The most expensive piece of the computation, the calculation of the document frequencies for all n-grams, is facilitated through the use of suffix arrays and preprocessing.

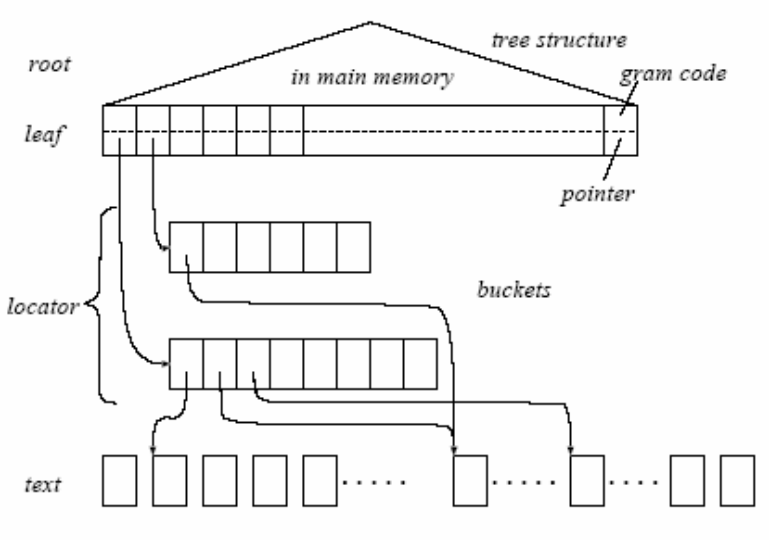


Figure 9: Gram-based index structure (cf. Sato et al. 2001).

[McNamee 2002] found that representing text using a combination of character n-grams of lengths one, two, and three is effective. A direct comparison with a basic bi-gram approach was, however, not carried out.
[Savoy 2004] discarded the most frequent bi-grams before indexing in order to get rid of noisy n-grams and reduce the size of the index.

Enhanced word-based approaches:

[Ogawa & Matsuda 1997] suggest an overlapping statistical word indexing method, which extracts some overlapping segments. In order to achieve better retrieval effectiveness with a smaller index, they modify their basic statistical word indexing method in two ways. First, the segmentation threshold is set below the optimal value, so as to prevent compound words from remaining unsegmented. This step extracts the "basic segments". In a second step, basic segments are merged into a larger segment by means of a newly introduced merge threshold. This second merging step limits the increase in index size caused by the lowering of the threshold. In addition, it leads to longer, possibly more specific compound phrases.

[Fujita 1999] focused on noun phrase indexing and its weighting issues. Phrasal terms were extracted in addition to single word terms contained in phrases and the term coefficient for phrasal terms against single word-terms was downweighted. This method outperformed single word indexing with long queries, while single-word-only indexing performed slightly better with short queries.

[Matsumura et al. 1999] proposed a Structured Index using dependency relationships between words in a sentence. The Structured Index is represented by a binary tree, which is constructed through dependency analysis and compound noun analysis based

on a word bi-gram. It did not, however, significantly outperform the TF-IDF-based baseline system, due to problems in dependency analysis and the matching and scoring algorithm.

Combination-of-evidence approaches:

These approaches merge the result lists obtained with more than one index type. Consequently, they must accept a higher storage cost required for the storage of several indices.

[Jones et al. 1998] successfully used a combination-of-evidence technique combining word-based and character-based indexing: “Indexing using dictionary based morphological analysis and character strings are both shown to be individually effective, but marginally better in combination.”

[Sakai et al. 1999] mention that the NEAT system combines character-based matching and morpheme-based matching, in order to avoid matching problems caused by non-explicit word boundaries.

[Vines & Wilkinson 1999] tried several different indexing strategies, i.e. character-based, word-based (ChaSen), bi-gram with (unsegmented) English strings, and subsequently combined the two best approaches: words without English and bi-grams without English. The document score was calculated by the simple formula $sim_{new} = 0.5 \cdot sim_1 + 0.5 \cdot sim_2$. This combination-of-evidence approach produced a further 1.2% average improvement.

[Kang et al. 2004] employed a word-n-gram coupling strategy combining several ranked lists generated from words and n-gram indexes differentiating both retrieval models and expansion term selection schemes. However, since he relied on preliminary experiments with Korean documents to select the coupling strategy, the coupled run did not yield the highest performance among all Japanese runs.

Hybrid Indices:

This idea was first proposed by [Tsang et al. 1999] for Chinese information retrieval as an alternative to the merging of retrieval lists as described in the previous paragraph. Although it has only been employed in Chinese IR so far, the results should be at least partly transferable to Japanese IR.

[Chow et al. 2000] founded their approach on the conclusion that words are the preferred index terms, if there is no Out-of-vocabulary problem. Bi-gram indexing is free from the out-of-vocabulary problem, however, bi-grams have a high storage cost.

Therefore, neither bi-grams nor words are exhaustively indexed in the document, but bi-grams are extracted where the out-of-vocabulary problem is likely to occur, namely at points in the text where phrases or sentences are segmented into individual character sequences. This approach has two advantages, namely the reduction of the storage demand vis-à-vis a simple bi-gram approach and solving the out-of-vocabulary problem.
The hybrid term approach was tested with several IR models and consistently achieved better average precision than those using words. However, the hybrid index incurred a slightly higher storage overhead than the word-based index. In addition, the recall performance was found to be lower for hybrid word-indexing than for word-based indexing.

[Luk et al. 2001] found that the averaged precision of hybrid term indexing was about the same as the best precision obtained with a bi-gram index, but incurring less storage (about 61% of the bi-gram index size). The precision performance of hybrid term indexing was found to be consistently better than that of word indexing, even though their storage requirement is about the same.

2.2.5 Pronunciation-Based Indexing

There is one more type of indexing approach, which was previously employed before the introduction of double-byte processing on computers. Before kanji could be handled by a computer, information processing systems used the katakana syllabary (cf. section 1.1.1) to represent Japanese text phonetically. Information retrieval systems at that time worked with 読み (“yomi”)-, e.g. pronunciation-based indices.

As explained in section 1.1.3, the Japanese language is very rich in homophones. In written language, the ideographic kanji characters normally facilitate correct interpretation. A phonetic transcription of Japanese lacks this information and can therefore be very ambiguous at times. Whereas human readers may still be able to guess the meaning of ambiguous words from the context, an information retrieval system incurs many false drops, which results in heavy losses in precision.

The advantage of the pronunciation-based index is that it is not sensitive to orthographic variants, e.g. okurigana or kanji variants, as described in section 1.2. The pronunciation-based index has become obsolete since the introduction of double-byte character handling, but it might be valuable in combination with other kinds of indices, especially for the handling of orthographic varieties.

It is difficult to compare the results of different research groups and their systems, even when experiments are carried out with a standard test collection, such as the NTCIR

collection. System performance is determined by so many different factors – which are, moreover, not always completely independent from each other – that it is hard to isolate a single one.

It can be concluded that there is no single best indexing strategy for Japanese IR and that every strategy has its advantages and drawbacks. The choice of an indexing strategy therefore largely depends on the context of the system, such as the preference of recall or precision, the amount of storage space needed/available, the importance of calculating speed, further functionalities to be implemented, other strategies to be applied, etc.

2.3 Optimization Strategies

This chapter presents optimization strategies that are commonly employed or have been suggested in monolingual Japanese IR. The term optimization strategy refers to an extension of the system which is added to the basic retrieval functionality in order to improve retrieval effectiveness.

If these strategies are also used in English or Western language IR, possible differences in their implementation will be pointed out.

2.3.1 Removing Stopwords

A very simple, yet effective strategy for enhancing system performance is the use of a stoplist. A stoplist contains terms with a very high text frequency, but very little semantic value, such as articles, pronouns, conjunctions, etc. Discarding those terms helps to both reduce the index size and achieve higher precision.

In Japanese IR, grammatical or structural words are often written in hiragana and automatically discarded together with all other hiragana words. When segmenting with a morphological analyzer, not only the individual words, but also information about their syntactical role is provided. In general, only semantically rich word types, i.e. nouns, adjectives, and verbs, are kept for the index.

Therefore, in Japanese IR, the use of stoplist is not as common as in other languages. However, a number of participants of the NTCIR workshops report using a stoplist. With the high number of possible n-gram combinations in Japanese (cf. section 2.2.2), it is also useful to discard n-grams with little informational value (cf. Savoy 2004).

2.3.2 Query Modification and Relevance Feedback

Another frequently employed optimization technique is Query Expansion through Pseudo Relevance Feedback (PRF), also known as Blind Relevance Feedback (BRF). As no user interaction is permitted in the SLIR and CLIR tracks of NTCIR and CLEF, query modification must take place automatically without relevance assessment information from the user.

Query Expansion techniques can be classified into local and global techniques. In local PRF, instead of using a sample set of relevant documents, the top n documents in the initial ranked output are assumed to be relevant and salient terms are extracted. Global techniques make use of a thesaurus or word co-occurrence information for query expansion.

General Observations

[Chen et al. 2002] observe in their Overview of the NTCIR-3 workshop that “QE is a cutting-edge technique for good retrieval performance” and “9 out of 12 top runs use query expansion”. As a result, PRF was also widely employed in NTCIR-4. Most groups apply standard PRF techniques, i.e. the Rocchio method or Robertson’s probabilistic method [Kishida et al. 2004].

Due to its heuristic approach, PRF is an unreliable technique. It has commonly been found to increase average retrieval effectiveness, yet is known to hurt performance for approximately one-third of a given set of search requests [Sakai et al. 2000, Sakai et al. 2001]. [Moulinier 2004], using a Rocchio-like formula, even found that PRF is helpful for only about 50% of the queries. A detailed analysis reveals that individual queries are greatly affected by PRF, either positively or negatively.

Critical parameters identified for feedback procedures are:

- the number of documents to be used for feedback
- the function used to score terms
- the number of terms to be extracted for feedback
- the weighting of these additional terms

In general, the following effects can be observed:

When the initial query is long and rich enough in terminology, the improvement given by the automatic feedback is limited, although it never hurts the performance.

Furthermore, relevance feedback can lead to an improvement of 11% to 15% even when initial retrieval results are already at a good level (cf. Fujita 1999).

The former observation is confirmed by [Savoy 2004], who found the percentage of enhancement greater for short topics than for longer ones.

Term Reweighting

Query Expansion is part of the broader concept of Query Modification, which is comprised of two components: Term Reweighting and Modification and/or addition of search terms.

Reweighting of query terms is based on the distribution of these terms in the relevant and non-relevant documents retrieved in response to those queries. This technique increases the rank of documents containing the reweighted terms. Changing search terms facilitates the retrieval of relevant documents containing terms that do not match the original query, but terms closely related to it (cf. Harmann 1992). Term Reweighing and Query Expansion seem to work best in combination.

[Chen & Gey 2002] found that adjusting term weight slightly improved the retrieval performance, whereas Relevance Feedback improved the performance by 12.82% without adjusting weight, and by 16.17% with reweighting.

Word-based vs. n-gram-based query expansion

[Ogawa & Mano 2001] compared word-based and n-gram-based Query Expansion in terms of speed and retrieval effectiveness. Their system uses an n-gram-based index with a word-based ranking algorithm. Morphological analysis, therefore, only takes place during query processing.

Word-based Query Expansion requires morphological analysis to identify words in each of the retrieved documents. Unlike in query processing, however, the amount of text that needs to be morphologically analyzed is much larger, resulting in a long processing time. Moreover, obtaining necessary frequency statistics of each word is costly, since these statistics have to be calculated using the data in the n-gram index.

Using n-grams as expansion terms eliminates much of the computation required for the word-based method. Hence, faster retrieval is possible. On the other hand, since n-grams do not reflect semantics as words do, retrieval effectiveness may suffer.

The results show that both word-based and n-gram-based query expansion are quite effective with no significant difference between the two methods. However, n-gram Query Expansion is considerably faster (14 times as fast as the word-based method).

Enhanced PRF approaches

There are also several enhanced PRF approaches that have been proposed in the NTCIR context.

To enhance the reliability of PRF, [Sakai et al. 2001] propose Flexible Pseudo-Relevance Feedback (FPRF), in an attempt to estimate the best PRF parameter values, i.e. the optimal number of pseudo-relevant documents, and the optimal number of expansion terms. As the results were inconclusive, [Sakai et al. 2004] proposed two new methods for determining the optimal number of pseudo-relevant documents, Term Exhaustion and Selective Sampling.

In the Term Exhaustion approach, the initial ranked output is scanned from the top, examining the query terms contained in the retrieved documents. The process is stopped when the frequency of 'novel' query terms (i.e. those that were not in the previous documents) drops below a threshold value.

Selective Sampling is unlike any other flexible PRF method in that it does not necessarily treat the top P documents as pseudo-relevant, that is, it is able to skip documents. The motivation is that there may be similar (and therefore redundant) documents among the top P documents, and it may be better in such a case to go further down the list to look for more "novel" documents.

While the Term Exhaustion results were rather disappointing, flexible feedback based on Selective Sampling proved to be effective for the NTCIR-4 Japanese test collection, especially with the <TITLE> fields, i.e. short queries.

[Sakai et al. 2002] tested several methods for the enhancement of basic PRF, namely Document Re-ranking, Term Selection Enhancement, Kanji Overlap Promotion (KOP), and Ranked Output Combination.

Document Re-Ranking involves re-ranking the initial (pre-PRF) ranked output based on sentence-internal co-occurrence of the initial search terms. It did not, however, lead to a significant improvement of retrieval effectiveness. Term Selection Enhancement takes into consideration the rank of the document in the search result and seems to lead to a slight increase of precision. Kanji Overlap Promotion is motivated by the fact that many kanji words are akin to concise summaries. Yet, it did not show a significant effect on the retrieval result. Since most kanji words are polysemous, the effect of KOP is sometimes positive, sometimes negative. Ranked Output Combination combines the

ranked output from full-text and summary indices, the summary index using only the title and the first sentence of each document. The full-text/summary ranked output combinations proved effective.

[Murata et al. 2002] used locational information as a characteristic of newspaper articles and found that this approach was often effective.

Global Feedback Methods

In addition to local feedback methods, in which feedback information is gained from a selection of relevant and irrelevant documents, feedback for query modification can also be acquired globally from the co-occurrence of words and their dependency relationships.

In fact, [Harman 1992] notes that ideally, Query Expansion should be done using a thesaurus for looking up synonyms, broader terms, and other appropriate words. However, the manually constructed thesaurus needed for this is seldom available, especially since it should be adapted to the text domain in question. Using a general-language thesaurus for a technical domain involves the risk of incurring a dramatic loss in specificity. Therefore, many attempts have been made to automatically create one, e.g. through term-term associations or clustering techniques.

[Tseng et al. 2004] attempted to generate an automatic thesaurus based on term co-occurrence statistics in order to apply it to query expansion. Since the calculation of term co-occurrences is computationally expensive, they proposed a more efficient method based on co-occurrence in the same logical segments of a smaller text size, e.g. a sentence or paragraph. They subsequently compared the performance between expansion by using this automatic thesaurus (“global expansion”) and PRF (“local expansion”). Results showed almost no effect of global expansion, whereas PRF was able to boost retrieval effectiveness substantially.

[Kanazawa et al. 2001] experimented with the relevance superimposition model, a sort of clustering approach in which document vectors are modified based on the relevance of the documents. In combination with query expansion, this approach led to a 9% increase in retrieval effectiveness.

2.3.3 Decompounding

As demonstrated in section 2.2.1 about indexing strategies, it is not only difficult to define word boundaries in Japanese, but also to determine the appropriate level of granularity with regard to compound words. Just as in German, compounding is used very productively in the Japanese language. Whereas a compound word is highly

specific, its component terms may be too unspecific, thus making it to decide on the optimal representation in the index. Indexing the compound split up into its component words or indexing it in its original long form as well as in a split form brings about a weight increase of the compound term in contrast to individual terms.

[Tomlinson 2004] found in an analysis of the NTCIR-4 topics, that the compound word “アップルコンピュータ“ (Apple Computer) in topic 42, split into “アップル” (apple) and “コンピュータ” (computer), raised precision by 25 points, because some relevant documents did not use “コンピュータ“, but only used “アップル”, or sometimes the hyphenated form “アップル・コンピュータ“.

In topic 52, decompounding “皇太子妃” (Crown Princess) to “皇太子” (Crown Prince) and “妃” (Princess) led to a 24 point increase in precision. The success of decompounding is explained by the fact that at least one relevant document used the split form “皇太子・雅子妃” (“雅子” being the name of the princess, “Masako”) and another relevant document did not contain”妃”, but only “皇太子”. Additionally, decompounding doubled the weight, because each piece was almost as uncommon on its own as the compound term.

The overall results showed a modest increase in mean average precision. However, decompounding did not have a positive effect for every topic.

2.3.4 Spelling Correction

In section 1.2, it was noted that Japanese has a high frequency of orthographic varieties. One way to deal with orthographic variation is spelling correction, where a fuzzy match between a word pair is made possible through the calculation of the editing distance between them.

Such a method should be especially promising for the handling of katakana loan words.

The proportion of European loan words in Japanese vocabulary increased from 1.4% in 1891 to 8-10% in the 1960’s. The greatest and most rapidly growing portion of European loanwords is comprised of English loans, due to the dominance of English in new domains such as the car and computer industries (cf. Taylor & Taylor 1995:288). In transcribing European words into Japanese, the nearest Japanese sound substitute is chosen for any sound not available in Japanese: common substitutions include ‘r’ for /l/; ‘b’ for /v/; ‘ts’ for /t/; ‘s’ for the initial sound of “thin”; ‘z’ for the initial sound of “this”. Further, consonant clusters, such as “cl-“ and “–ld”, are broken up by inserting a vowel, and final consonants, such as ‘–m’ and ‘–l’, are converted into CV (C=consonant, V=vowel) pairs by attaching a vowel (cf. Taylor & Taylor 1995:289).

Due to the different and limited sound system in Japanese, transcription can only be an approximation of the original pronunciation. For uncommon or newly coined English words, there often coexist a number of different versions of transcriptions (cf. section 1.2.3).

Spelling correction provides a way to treat those slight variants as quasi-synonymous by assigning them a very high degree of similarity.

Of course, spelling correction can only work for word-based index terms. N-gram-based indices already incorporate a sort of spelling correction, as the comparison of n-grams can be regarded as a comparison of character subsequences of the original text string.

3 Approaches in Cross-Lingual Japanese/English IR

> *"The central challenge in cross-language information retrieval (CLIR) is to find the most effective way to bridge the language barrier between queries and documents. […] The optimal method of choice for a specific CLIR task, however, does not only depend on the technical strength of a method, but also depends on the availability and quality of the knowledge sources or parallel corpora that a CLIR system can employ for the right domain."*
>
> (cf. Yang & Ma 2002)

This section will first present methods to overcome the language barrier between queries and documents. In the next step, established translation approaches will be introduced. and some of the translation resources for Japanese and English will be presented. Finally, strategies for the optimization of the translation step are described.

[Sakai et al. 1999] report that CLIR involving Japanese is more problematic than CLIR between European languages, one problem being once again the term-selection step with the hurdle of word segmentation.

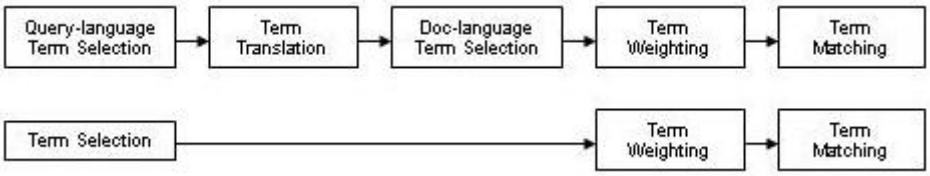


Figure 10: CLIR vs. monolingual IR (Oard & Wang 1999).

As the above figure shows, the first step in both mono- and cross-lingual retrieval is term selection. When processing European languages, tokenization can easily be achieved using white spaces as word delimiters. The only difficulties may be phrase recognition and, for languages like German, compound splitting. In Asian languages, however, segmentation itself represents a major hurdle. As incorrectly segmented terms cannot be translated correctly, this has direct consequences for the retrieval performance.

Further difficulties in Japanese-English CLIR are the lack of a linguistic relation between the two languages and their different character sets. With European languages, cross-lingual exact string matches are sometimes attempted when no translation is known for a word, e.g. in the case of proper names.

3.1 Translation Strategies

The first decision that has to be made when performing cross-lingual IR is to determine the manner in which way documents and queries should be matched. [Oard & Wang 1999] identify four fundamental ways to match queries in one language with documents in another. These are depicted on the left side of Figure 11. Closely related to the choice of translation strategy is the choice of translation resources, showed on the right side of the figure.

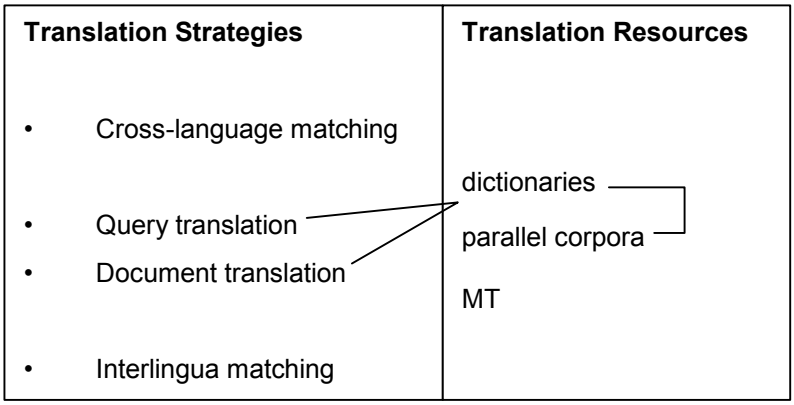


Figure 11: Translation strategies and translation resources.

3.1.1 Cross-Language Matching

The cross-language matching approach leaves queries and documents untranslated. Translation knowledge is embedded in the matching algorithm itself. This is only practicable for languages that are sufficiently close to each other, such as Roman languages (cf. Buckley et al. 1998). In European IR, cross-language matching is an interesting solution for the handling of proper names, e.g. “Roma”, “Rome”, “Rom”. However, it is not suitable for English/Japanese CLIR, where no rules for a direct transformation from a word in the source language to the corresponding word in the target language can be established. A variant of cross-language matching might prove useful in a special case: transliteration of Japanese katakana terms to their English equivalents (cf. section 3.4.2).

3.1.2 Query- vs. Document Translation

Both query and document translation reduce the cross-lingual retrieval task to a monolingual one after translation.

In the NTCIR-4 workshop, most groups adopted the query translation approach (cf. Kishida et al. 2004). This is due to the fact that this approach is the most efficient one: it only requires the translation of a small amount of text and can be done quite quickly and inexpensively. A dictionary-based query translation (term-by-term translation using a term list) is easily implemented and is well-known to produce about half the retrieval effectiveness of monolingual systems. “A query translation module and existing monolingual retrieval engines [can be combined] with minimal cost” (cf. Fujii & Ishikawa 1999). However, one major shortcoming of the query translation approach is that queries submitted by ordinary users are usually very short and consist of just an enumeration of keywords, i.e. no context provided. Especially with a small number of search terms, it can be fatal not to be able to translate these correctly.

The approach based on translation of target documents, on the contrary, can exploit the context information of a whole document for disambiguation. Existing machine translation systems, if available, can easily be utilized for this task (cf. Kimura et al. 2004).

Therefore, by and large, document translation approaches achieve better retrieval effectiveness than those based on query translation (cf. Sakai 1999, Sakai 2000). However, the document translation approach is applicable only if the size of the document collection is reasonably small, or alternatively, if the language familiar to the user is known in advance so that the translation can be done offline.

[McCarley 1999] compared document and query translation for CLIR between English and French and found that hybrids of document and query translation-based systems (the score of a document being the arithmetic mean of its scores in the query and document translation systems) outperform query translation systems, and even human-quality query translation systems. He also demonstrated, however, that a direct comparison between document and query translation is not straightforward, because two different translation systems must be involved.

Based on McCarley’s findings and seeking to achieve the high translation quality of document translation systems without losing too much computational speed, [Fujii & Ishikawa 2000] proposed a two-stage method, which integrates the query and document translation methods.

In the first stage, the query translation method is used to retrieve a limited number of foreign documents. This limited number of documents is then machine-translated into the query language in the second stage, thus minimizing computational cost. Finally, the translated documents are re-ranked based on the score, combining the documents individually obtained with query and document translation methods. Preliminary experiments with the NTCIR-1 Japanese-to-English CLIR collection showed that the two-stage method outperformed the query translation method.

[Gey 2004] proposed the “fast document translation” method, involving a “surface” word-to-word translation of the corpus using a simple bilingual lexicon. The lexicon is constructed collecting unique words from the corpus and submitting each individually to the translation engine in order to obtain a unique word in the topic language. Although only less than half of the words in the corpus could be translated into English, the run yielded satisfactory results.

3.1.3 Interlingua Matching

Interlingua approaches use semantic annotation (by way of thesauri or ontologies) or Latent Semantic Indexing (LSI). LSI examines the similarity of the contexts in which words appear and creates a reduced-dimension feature space, in which words occurring in similar contexts are placed in close proximity to one another. In this way, term-term interrelationships, which are disregarded by Boolean retrieval, are automatically modeled and can improve retrieval performance.

There is no need for linguistic resources such as dictionaries or thesauri to determine these word associations, which are instead derived from a numerical analysis of existing texts. The learned associations are specific to the domain of interest, and are produced completely automatically.

[Dumais et al. 1997] proposed a cross-language LSI method, in which a parallel bilingual learning corpus is generated from an initial sample of translated documents. An LSI analysis of the training documents results in a dual-language semantic space in which terms from both languages are represented. Standard mono-lingual documents are then “folded in’’ to this space on the basis of their constituent terms. Queries in either language can retrieve documents in either language without the need to translate the query, because all documents are represented as language-independent numerical vectors in the same LSI space and “translation” is carried out by matrix computations.

[Jiang & Littmann 2001] tested three different vector-based (cross-lingual) retrieval methods: LSI, local LSI, and Approximate Dimension Equalization (ADE), using the NTCIR-1 collection. Although ADE and local LSI proved very effective and were comparable in performance with the best systems, subsequent experiments using the NTCIR-2 collection did not reproduce comparably satisfying results. This might be due to the fact that training with the NTCIR-1 corpus was not appropriate for retrieval with the NTCIR-2 collection.

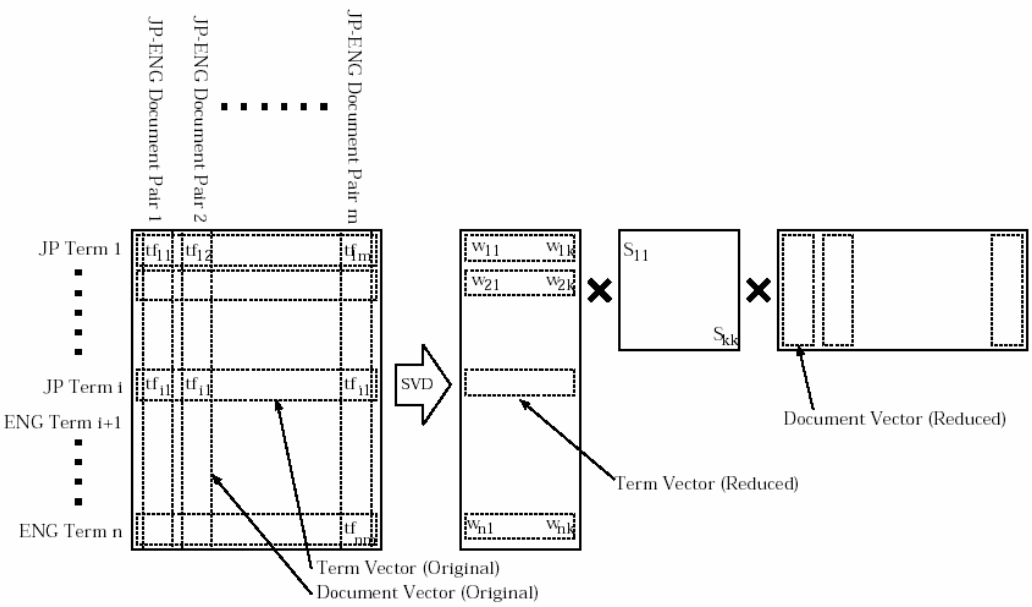


Figure 12: Cross-Language Latent Semantic Indexing (cf. Mori et al. 2001).

The two main problems with CL-LSI are unknown words and words which appear only in some sub-corpora. Words not appearing in the set of dual-language documents are completely ignored, because no translation information can be obtained for them.

3.2 Translation Resources

There are three basic translation resources: dictionaries, MT systems, and parallel corpora (cf. Figure 11).

The choice of which translation resource to use depends not only on its performance, but also on its availability. High quality MT systems are often expensive and parallel corpora are labor-intensive to create.

3.2.1 MT-System-Based Translation

The top English-Japanese runs in NTCIR-2 and 3 used MT translation (TSB group with Toshiba MT). The system employs the transfer method with a multi-stage disambiguation mechanism. In follow-up experiments, average performance could be improved considerably by including the synonyms from the system's final disambiguation stage [Sakai et al. 2002].

However, good MT systems are the most expensive type of resource, which is why they are not often employed by NTCIR participants in spite of their good performance.

3.2.2 Dictionary-Based Translation

Using a simple dictionary to look up translation terms is computationally much less expensive than having a whole document translated by an MT system. However, word-to-word translation is not always appropriate, especially for compound words, which are usually not found in a bilingual dictionary. A possible solution is the translation of their individual components and the subsequent disambiguation of possible combined translations. Ambiguity can be solved by building domain-specific bilingual dictionaries, assuming that a compound word translates into one translation in a domain (cf. Tanimura et al. 2001).

[Sawada & Umemura 1999] propose a dynamic programming method to aggregate similarity. They argue that this method should be especially useful for Japanese-English IR, where performance suffers from word-sense ambiguity and the difficulty of capturing appropriate phrases, especially when dealing with documents where technical terms play an important role. *"IR systems usually regard a document as a set of words. This assumes that the words are usually [an] efficient handle to retrieve documents. This is not always the case for cross-lingual IR. A word in a language may correspond to several words in another language. This will degrade the precision of [the] IR system. In addition to this, an important but new technical term may not exist in the dictionary. This may make the system fail to retrieve the document. Moreover, appropriate word boundaries may not be apparent in some languages."* (ibid.).

They developed their method based on the observation that a technical term in English usually consists of several words and that the corresponding Japanese technical term usually preserves the order (this was true for 1,943 pairs of terms out of 2,000 by random sampling from a technical dictionary).

By regarding the replacement of a word with a corresponding English word as one edit operation, the cross-lingual distance between two sequences of words can be defined using this operation with appropriate weights. This makes it possible to accommodate sequences of words, rather than just one word. The optimal translation is calculated by the dynamic programming method. The DP (Dynamic Programming) system attempts to translate every substring of the given query and to find the best way to accumulate the score, preserving the order of the words.

[Collier et al. 1998] compared the performance of dictionary-lookup vs. machine translation, applied to aligning 1,488 news articles written in Japanese to 6,782 articles written in English. Results show that translation by dictionary lookup performs as well as machine translation.

A great difficulty in dictionary-based translation is the choice of translation terms or translation disambiguation. Most of the words in dictionaries are associated with several possible translations, which need to be disambiguated in order to avoid an unintended shift in meaning. Choosing more than one translation candidate may have a query expansion effect, if synonyms or the source language term are integrated into the query. The challenge is to find the optimal number of translation terms without losing concept specificity.

How many translation terms should be retained? How should those terms be selected? Possible solutions are: “select-first” (selecting the first translation candidate only), “select-best” (based on word co-occurrence in a corpus), “select-all” (selection of all translation candidates). In a study by [Lin et al. 1999], “select-first” proved to be the best-performing strategy, followed by “select-best”.

[Oard & Wang 1999] tested two automatic dictionary-based query translation techniques with four variants of the queries and found that longer queries yield a better performance, and that the use of the first translation in the EDICT dictionary[30] (cf. section 3.3.1) is comparable with the use of every translation. Japanese query term segmentation was carried out with the morphological analyzer JUMAN and posed no unusual problems.

Another strategy for the disambiguation of translation terms is the combination of a dictionary-based translation with the use of a parallel corpus for translation disambiguation. This shall be discussed in greater detail in section 3.2.4.

3.2.3 Corpus-Based Translation

Corpus-based translation is based on aligned parallel corpora. Translation is determined by the computation of the association strength between the Japanese word and every English word co-occurring with the original term in at least one aligned sentence pair. The alignment can be achieved with the sentence alignment technique proposed by Gale & Church (cf. Chen et al. 1999). The suggested translations are ranked according to their association strength. The number of translation terms to be retained is heuristically determined, based on the type of Japanese word, i.e. kanji or katakana, and the number of characters in the Japanese word.

[Sato et al. 1999] achieved 68-79% in average precision, and 60-73% in R-precision, compared to the monolingual baseline. The recall of the CLIR runs was 14-15% higher

[30] 64,433 Japanese entries and a total of 104,705 bilingual term pairs

than monolingual retrieval. This might be due to a certain query expansion effect of the translation step. The disadvantage of the corpus-based method is its complete dependence on the parallel corpora – in quality, quantity, and in the domain covered. Terms which do not occur in the parallel corpora cannot be translated.

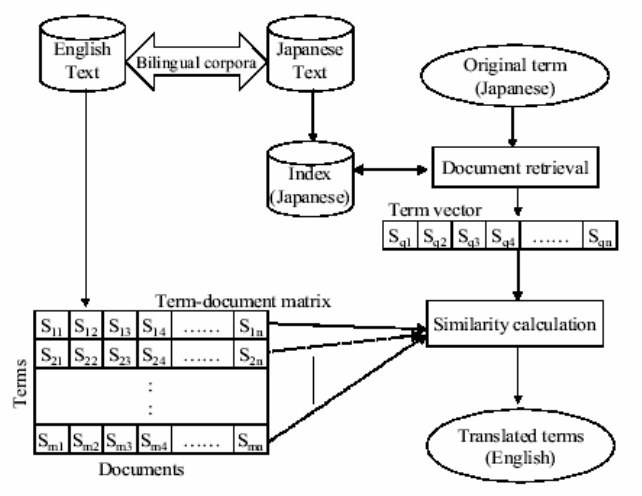


Figure 13: Translation term selection using similarity calculation (cf. Sato et al. 1999)

[Nakazawa et al. 1999] proposed a translation disambiguation method based on comparable corpora, which are more readily available than parallel corpora. They first expand the source language query terms and then translate them stepwise using a bilingual dictionary and the GDMAX method to calculate the term co-occurrence frequency in a bilingual dictionary.

[Fujita 2001] tested the effectiveness of parallel corpus usage, instead of a classical translation step, for Japanese-English CLIR with the NTCIR-1 collection. They first carried out a search in the source language and obtained the n top-ranking documents. Their counterparts in the target language were used for the extraction of target language search terms. This strategy worked extremely well, reaching 36.80% average precision. This excellent performance could however not be reproduced with the NTCIR-2 test collection, where only 25% of the documents have parallel equivalents (compared to 55% in the NTCIR-1 test collection), which shows the dependence of this method on the given resources.

[Nakagawa & Kitamura 2004] obtained high performance with both a parallel corpus and an MT system. The highest performance was yielded with a combination of all resources.

3.2.4 Combined Dictionary- and Corpus-Based Approaches

It is a popular strategy to combine dictionary- and corpus-based approaches. While a dictionary provides translation candidates, a parallel corpus can be used to derive co-occurrence statistics to deal with translation ambiguity (cf. Lin et al. 1999). Mutual information is used to measure the degree of correlation between two words.

The co-occurrence information is trained from a monolingual corpus, which means that neither a bi-lingual nor an aligned corpus is needed. However, the parallel corpus should be very large and balanced in order to provide correct disambiguation information (cf. Lin et al. 1999).

[Chen et al. 1999], for example, segment the text to be translated by a dictionary-based longest-matching technique and retain, for each word, the most frequent English translation in the NTCIR-1 collection.

The World Wide Web may also be used as a corpus for translation disambiguation. [Zhou et al. 2004] counted the number of Web pages including a pair of translation candidates and [Kimura et al. 2004] extracted disambiguation information from Web documents within a Web category corresponding to the query.

[Sato & Noguchi 2001] carried out experiments comparing a pure corpus-based approach, a pure dictionary-based approach, and a combined corpus- and dictionary-based approach. The dictionary was constructed merging the EDICT and some other bilingual resources. In the hybrid method, the bilingual dictionary was used where terms that did not occur in the corpus could not be translated. About 91% of Japanese words had less than 3 English translation words. The remaining 9% of words should be disambiguated.

Results showed that corpus-based translation achieved higher performance than the dictionary-based method in both short and long query runs. Moreover, the combination of the dictionary- and the corpus-based method led to an improvement in both precision and recall. The combined corpus-based and dictionary method achieved 66% average precision and 110% recall compared to the monolingual baseline.

3.3 Overview of Selected Translation Resources

3.3.1 Dictionaries

There are a number of free dictionaries available on the Internet for translation to and from Japanese. The following will provide a short survey of a selection of commonly used resources:

EDR bilingual dictionary[31]

The EDR Electronic Dictionary was developed for advanced processing of natural language by computers and is composed of eleven sub-dictionaries. These include word dictionaries (English and Japanese), a bilingual dictionary (J-E with about 230,000 entries and E-J with about 160,000 entries), a concept dictionary, a co-occurrence dictionary, and a technical terminology dictionary, as well as the EDR corpus. A usage fee has to be paid for scientific use.

EDICT[32]

EDICT is the major freeware Japanese-English lexicon developed by Jim Breen et al. and can be used online via the WWWJDICT server[33]. The original EDICT file has approximately 110,000 entries. Further, there is a number of domain-specific adjuncts or spin-offs:

- COMPDIC - Computing & Telecommunications terms (over 12,000 entries)
- ENAMDICT - a large file of Japanese place and person names (over 350,000 entries)
- LIFSCIDIC - the Life Sciences dictionary of bio-medical terms
- JDDICT - a short Japanese - German dictionary
- FINMKTDIC - financial and marketing terms
- MISCDIC – a collection of small glossary files covering aviation, law, geology, concrete, pulp & paper, etc.

[31] http://www.jsa.co.jp/EDR/
[32] http://www.csse.monash.edu.au/~jwb-edict.html
[33] http://www.csse.monash.edu.au/~jwb/wwwjdic.html

- J_PLACES - a file of Japanese place-names derived from the extensive postal-code database (under construction)
- THE_LOT - a combination of the EDICT, COMPDIC, ENAMDICT, LIFSCIDIC, FINMKTDIC, and MISCDIC files. Useful for text glossing and wide-ranging searches.

JMDict[34]

The JMDict (Japanese-Multilingual Dictionary) is compiled and maintained by Jim Breen and The Electronic Dictionary Research and Development Group at Monash University, Australia. The project has as its aim the compilation of a multilingual lexical database with Japanese as the pivot language. The project began in 1999 as an offshoot of the EDICT Japanese-English Electronic Dictionary project. It involved a major overhaul of the main files, with a more complex structure using XML. The most recent release was in May 2005. JMDict has reached a size of approximately 100,000 entries, with most entries having translations in English, French (approx. 58,000) and German (approx. 83,500, adapted from the WaDokuJT Project described in the next section).

Further, there are 4,800 entries with Russian translations, and a set of approximately 4,500 Spanish translations is being prepared, with the prospect that some 20,000 will be available shortly (cf. Breen 2004).

WaDokuJT[35]

WaDokuJT is a comprehensive German-Japanese Dictionary developed by Ulrich Apel using EDICT and ENAMDICT data. The current version has 211,300 entries, plus 49,000 variants with 87,000 lemmata.

It is also possible is to build a bilingual dictionary from scratch. Most methods adopt a parallel bilingual corpus, i.e. documents and their translations, and carry out alignment. The advantage is that the resulting dictionary is very domain specific. The problem is finding a sufficiently large parallel corpus. In the first NTCIR workshop, a considerably large part of the corpus was bilingual. This fact was exploited by several groups for the construction of a dictionary by aligning the English and Japanese keyword fields (<KYWD> and <KYWE> fields with 1,439,992 entries), e.g. [Fujita 2001], [Chen et al. 1999]. This is a simple and effective method, as confirmed by the results. However, this

[34] http://www.csse.monash.edu.au/~jwb/j_jmdict.html
[35] http://www.wadoku.de/, downloadable at http://bunmei7.hus.osaka-u.ac.jp/download.htm

method can only be applied when the documents containing keywords in both the source and target languages are available for creation of a bilingual dictionary. Moreover, the same dictionary, however, is not useful for the current NTCIR document collection, as its focus was on scientific texts. If a system must be flexible and able to cope with queries in many different kinds of topics, such as in Web IR, it is also not practical to prepare corpora for all possible domains.

3.3.2 MT Systems

The most commonly used and freely available MT systems for English and Japanese are:

- BabelFish MT system[36]
  Online text and web page language translation service provided by altavista.
- YakushiteNet MT system[37]
  Japanese online text and web page language translation service.
- GOOGLE Language Tools[38]
  Beta version of online text and web page Japanese/English translation.

[Savoy 2004] compared several different translation resources, i.e. the MT systems BabelFish[39], FreeTranslation[40], InterTran[41], and WorldLingo[42], and the machine-readable dictionaries EvDict[43] and Babylon[44]. The BabelFish MT system produced the best results for Japanese, but a combination of WorldLingo and the Babylon dictionary produced an even better MAP value.

[36] http://babelfish.altavista.com/
[37] http://yakushite.net/
[38] http://www.google.com/language_tools?hl=en
[39] See [36].
[40] http://www.freetranslation.com
[41] http://www.tranexp.com:2000/InterTran
[42] http://www.worldlingo.com
[43] http://www.samlight.com/ev
[44] http://www.babylon.com

3.4 Translation Optimization Strategies

3.4.1 Web Resources for the Translation of OOV Terms

The out-of-vocabulary problem mentioned in the context of word segmentation is equally crucial for the translation process, especially in the case of CLIR with language pairs possessing few inter-language cognates, such as Japanese and English (cf. Qu et al. 2004). Terms that cannot be translated are lost for the querying step. Methods for solving the OOV problem are therefore vital.

In the NTCIR-4 workshop, several groups used Web resources in order to obtain translation information for words not contained in the dictionaries.

[Seo et al. 2004] manually collected translation information of unknown words (especially proper names) from the Web and thus enriched their bilingual Korean/English dictionaries.

[Kwok et al. 2004] used Web resources for an automatic extraction of translations for unknown words. They exploited the fact that translations in bilingual documents tend to be expressed in the following form:..aaaa (bbbb).. or ..bbbb (aaaa).., where aaaa and bbbb are text snippets of language a and b, respectively. Unknown terms are therefore submitted to a Web search engine and the results are searched for the above pattern in order to obtain a translation. The method was successfully applied to English-Korean CLIR, and it should function for the English/Japanese language pair as well.

A similar approach was adopted by [Zhang & Vines 2004]. They developed a sophisticated Web mining algorithm for identifying translations of unknown Chinese words using the Google search engine and co-occurrence statistics for extracting English equivalents from Chinese Web documents.

[Chen & Gey 2002] developed a procedure for automatically extracting Japanese translations of English words from search results returned from Internet search engines using English words as queries. Babelfish was used for translation of English topics into Japanese. Assuming that the untranslated English words or phrases are mostly proper nouns, including personal names, those were not further looked up in other translation systems or bilingual dictionaries, but were submitted to Yahoo!Japan. Up to 200 search result entries were downloaded and the result entries were then segmented into words using ChaSen. The Japanese words surrounding the English word were ranked and the two top-ranked “translations” were used to replace the untranslated English words. The method achieved a 38.23% increase in average precision, which is a substantial improvement compared to a pure Babelfish translation. However, in three cases precision was zero or near zero.

3.4.2 Transliteration

Another promising technique for the handling of a special group of OOV terms, English loan words represented in katakana, is transliteration.

Through transliteration, out-of-dictionary query terms can be automatically associated with phonetic equivalents in a target language. This method is effective in translating words imported from a foreign language and spelled out by phonetic alphabets, such as katakana in Japanese. Japanese usually represents loanwords (primarily for technical terms and proper nouns) based on the katakana syllabary. These can never be exhaustively enumerated in a dictionary and are therefore prone to be OOV terms.

When representing English or foreign words in the phonetic katakana syllabary, the Japanese choose the phonetically closest correspondent to replace the foreign sound. In most cases, this process is rather regular. Table 12 shows how foreign words are decomposed and the components transformed into katakana syllables.

English	*katakana*
system	*shi-su-te-mu*
mining	*ma-i-ni-n-gu*
data	*dee-ta*
network	*ne-tto-waa-ku*
text	*te-ki-su-to*
collocation	*ko-ro-ke-i-sho-n*

Table 12: Examples of English-katakana correspondence (cf. Fujii & Ishikawa 1999).

For the automatic creation of a dictionary through transliteration, there are several difficulties which arise from the particular ways of coining loan words in Japanese. [Taylor & Taylor 1995:290] identify five major kinds of European loan words:

1. Words that represent European objects and concepts:
   - banana
   - arufabetto (‘alphabet’)
2. Words that represent objects and concepts that have native words:
   - risuto (‘list’)
   - ruutsu (‘root’)
3. European words truncated:
   - masukomi (‘mass communication’)
   - waapuro (‘word processor’)
   - pasokon (‘personal computer’)
   - mazaakon (‘mother complex’)

- rimokon (‘remote control’)
- eakon (‘air conditioner’)
- sekuhara (‘sexual harassment’)

4. European words somewhat changed in meaning:
   - haikara (‘high collar’ → ‘modish’)
   - waishatsu (‘white shirt’ → ‘dress shirt of any color’)
   - abekku (‘avec’ French ‘with’ → boy-girl dating’)
   - macho (‘macho’ → ‘buff, muscular’)
5. New words coined from existing European words:
   - ooeru (‘OL’ for ‘office lady’)
   - oorudomisu (‘old miss’ for ‘spinster’)

Only the first group does not represent any difficulty for transliteration techniques, provided that the original word is an English one. In the case of the second group of loan words, those with native Japanese counterparts, the desirable translation might be the Japanese and not the foreign one. The third group, truncated loan words, represents a serious problem for transliteration algorithms. Sometimes truncation makes it almost impossible to recognize the original.

The fourth group, loan words whose meaning has changed with the adoption into the Japanese language, can lead to a shift in meaning when translated to their original roots. Similarly, newly coined words cannot be traced back to their origin.

The classical transliteration technique for the transliteration of a word written in English into the corresponding word in Japanese was developed by [Knight & Graehl 1998]. They built a stochastic model of phoneme translation using 8,000 pairs of words in English and borrowed words in Japanese. Since the method requires a phoneme inventory, it cannot generate English words whose phonemes are not listed. It further assumes that an English word always translates into a borrowed word in Japanese. As mentioned above, however, cases are frequent where an English word translates not into a borrowed word, but a native word in Japanese.

A very thorough analysis of query translation difficulties for Japanese-English IR and strategies to cope with the OOV problem was carried out by [Fujii & Ishikawa 1999]. They combine three dictionaries (cf. Figure 14), which are consulted sequentially until a translation term can be found. The transliteration dictionary is used only for katakana base words.

After deriving possible translations for base words, translation ambiguity is resolved using a probabilistic model.

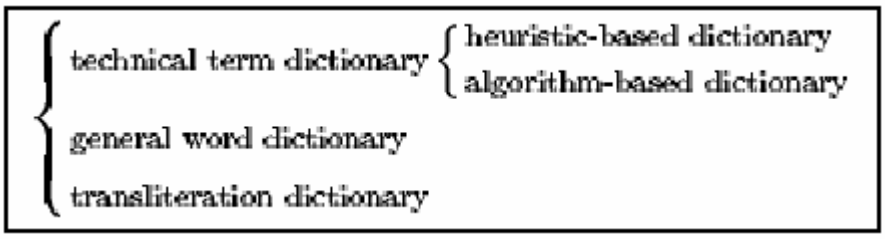


Figure 14: Combination of dictionaries used by [Fujii & Ishikawa 1999].

The technical term dictionary used was the EDR technical terminology dictionary which includes 120,000 English-Japanese translations related to the information processing field.

For transliteration, the approximate similarity between romanized katakana syllables and English syllables is defined (cf. Table 13) and a shortest-path algorithm is used in order to define the closest word pair (cf. Figure 15). A measure of distance obtained from the transliteration of an English word into a Japanese word is used in order to avoid transliterating a word with a native Japanese equivalent.

Condition	Similarity
e and *j* are identical	3
e and *j* are phonetically similar	2
Both *e* and *j* are vowels or consonants	1
Otherwise	0

Table 13: Similarity measure between English and Japanese characters (cf. Fujii & Ishikawa 1999).

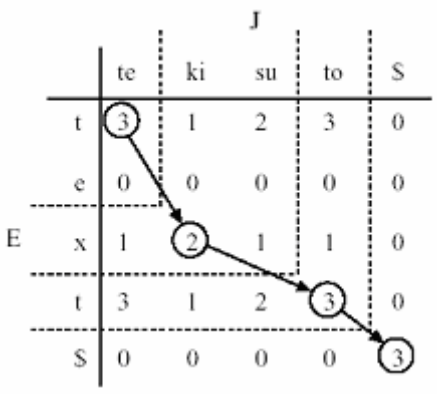


Figure 15: An example matrix of English-Japanese symbol matching (cf. Fujii & Ishikawa 1999).

The combination of technical term and general word dictionaries is intended to deal with technical compound words containing general base words (e.g. “AI shougi” – AI

chess game). The Algorithm-based dictionary performed best (Basic assumption: the set of technical terms in a given domain should consist of as few base words as possible).

It was found that the performance was improved with increasing number of dictionaries used.

[Fujii & Ishikawa 2004] claim that a method to resolve back-romanization- ambiguity is needed. They were able to trace back the poor performance of the English-Japanese topics to Romanized Japanese person names, such as “Akira Kurosawa”. So far, their transliteration method can only be applied to katakana words.

3.4.3 Pre- and Post-Translation Expansion

> *“Pseudo-relevance feedback, while useful in monolingual applications for refining and enriching short user queries, proves even more important in cross-language information retrieval (CLIR). […] For CLIR, query expansion before and after translation can provide an opportunity to recover from translation gaps, reduce ambiguity, and enhance recall.”*
>
> (cf. Levow 2004)

It is expected that pre-translation PRF picks up related terms of original query terms before the translation process, whereas post-translation PRF helps to absorb the noise introduced by the translation step.

A study by [Sakai 2000] confirmed that PRF is more effective for CLIR than for monolingual IR. This might be due to the fact that PRF in general shows greater effect for queries with moderate initial performance. With post-translation expansion, [Sakai 2000] achieved a Japanese-English CLIR performance comparable to the best-case monolingual IR results.

[Levow 2003] claimed that for CLIR across languages where different orthographies or character encodings prevented cognate matching, post-translation expansion in document and query translation architectures respectively played an integral role as a means of recovering crucial and often untranslatable Named Entities. She based her statement on experiences gained in CLIR with Chinese, but the same should be true for Japanese, where Named Entities also represent a special challenge for IR (cf. section 3.4.5).

[Levow 2004] compared pre- and post-translation query expansion, as well as their combination, and found an improvement over the unexpanded baseline for all three strategies, with large and highly significant improvements for the combination of both pre- and post-translation expansion.

[Fujita 2001] tested pre- and post-translation query expansion, executing a pilot search against the source language database before and after the query translation. Pre-translation PRF was always effective and clearly helped to compensate for information lost in the translation step. An improvement of as much as 16.5% could be obtained using this method.

3.4.4 Phrasal Translation

Phrasal translation refers to the utilization of noun phrases extracted by linguistic processing as translation units. [Ballesteros & Croft 1997] suggested that phrasal translation can greatly improve effectiveness, but improvements are more sensitive to the quality of the translations than single words. One poor translation can counteract any improvement gained by the correct translation of several phrases.

[Fujita 2001] found that the translation process using single terms introduced many noisy terms or wrongly translated terms. These were mainly frequently occurring single-word terms with low TDF values. Phrases, on the contrary, are normally very good translation units. He therefore claims that phrasal translation is crucial for better query translation, irrespective of the usage of phrasal terms in the retrieval engine itself. The translated phrase may be used as decomposed single words for the sake of robustness.

3.4.5 Named Entities

Named Entities (NEs) in general represent a difficulty for cross-lingual IR, especially when they mark "domestic topics", in cases where the NEs are known or of importance only in one location or language. When an NE is adopted in another country, the pronunciation and orthography are often altered to some degree. Changes are particularly salient when a transcription into another writing system or script is involved. Japanese proper names which have been transliterated to the English script, cannot easily be back-translated to the Japanese script (cf. 3.4.2). A Japan-specific problem with people’s names is the convention of stating family name first, followed by the first name. In an international context, the Japanese sometimes, but not always, adapt to foreign conventions and change this practice. This further complicates name identification.

4 System Overview

4.1 The MIMOR Framework

4.1.1 Basic Assumptions

The MIMOR model was originally inspired by the main outcomes of TREC, where it was found that many IR systems perform similarly well in terms of recall and precision but do not lead to the same sets of documents. Multiple indexing and fusion approaches try to profit from these findings in order to gain access to a greater share of relevant documents through the integration of several techniques.

On the other hand, relevance feedback is a very promising strategy for improving retrieval quality.

MIMOR represents an information retrieval system managing poly-representation of queries and documents by selecting appropriate methods for indexing and matching (cf. Mandl & Womser-Hacker 2001). By learning from user feedback on the relevance of documents, MIMOR learns which combinations of object representations and IR functionality lead to good performance of the overall system. An internal evaluation procedure, which is realized via a blackboard model, permanently registers which resource produces good results and which one does not. Well-performing techniques gain high weights; poorly-performing ones are excluded over time.

4.1.2 Modelling of Fusion and Learning

From a computational point of view, MIMOR is designed as a linear combination of the results of different retrieval systems. The contribution of each system or algorithm to the fusion result is governed by a weight for that system.

$$RSV_{MIMOR}(doc_i) = \frac{\sum_{system=1}^{N} (\omega_{system} RSV_{system}(doc_i))}{N}$$

Equation 1: Calculation of the Retrieval Status Value in MIMOR (cf. Schneider et al. 2004).

Different retrieval systems can be completely different retrieval engines using different IR models, as well as variations of one system, e.g. the same retrieval engine, operating on an n-gram and a word-based index.

A central aspect in MIMOR is learning. The weight of the linear combination of each IR system is adapted according to the success of the system measured by the relevance feedback of the users. A system which assigned a high retrieval status value (RSV) and consequently a high rank to a document which received positive relevance feedback should contribute to the final result with a higher weight. The following formula enables such a learning process, also illustrated in Figure 16:

$$\omega_{system} = \varepsilon \; RF_{user}(doc_i) \; RSV_{system}(doc_i)$$
$$\varepsilon \quad learning\ rate$$

Equation 2: Formalization of the MIMOR learning process (cf. Womser-Hacker 2005).

However, the optimal combination may depend on the context and especially on the users' individual perspectives, as well as the characteristics of the documents. Therefore, MIMOR needs to consider context.

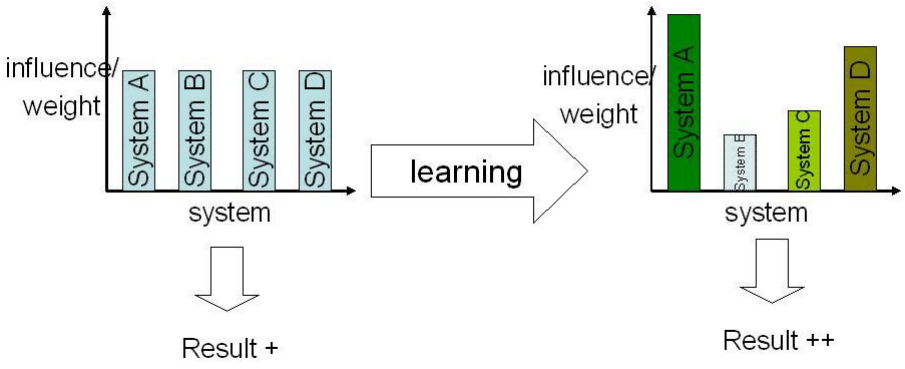


Figure 16: Learning the optimal linear combination over time (cf. Womser-Hacker 2005).

The performance of IR systems differs from domain to domain. Within the context of the TREC, it was found that particular document characteristics relevant for the indexing procedure may be responsible for this effect. In one experiment, for example, optimal similarity functions, especially for short queries, could be developed (cf. Kwok & Chan 1998). MIMOR is based upon the idea that formal properties of queries and documents can be exploited in order to improve the overall fusion system. Within fusion,

the weight of a system should be high only for the type of documents for which it was optimised (cf. Mandl & Womser-Hacker 2001).

4.2 The Lucene Search Engine Technology

Lucene[45] is an open-source IR library, written in Java and licensed under the Apache Software license. It was originally created by Doug Cutting and is now being developed by a team of about half a dozen active programmers.

The primary advantages of Lucene are performance, scalability, and extensive adoption (cf. Gospodnetić & Hatcher 2004:311). A fairly large and active user community is constantly contributing add-on open source code for various specialized IR tasks. The following section will describe Lucene's basic architecture and selected extended features, which are relevant for the implementation of Japanese language support. This section is drawn from the book Lucene in Action by [Gospodnetić & Hatcher 2004]

4.2.1 Architecture

Lucene is not a full-featured search application, but rather an IR toolkit. It provides a powerful core API (Advanced Programmer's Interface) that can be integrated into other programs and/or customized at will.

[45] http://jakarta.apache.org/lucene/docs/index.html

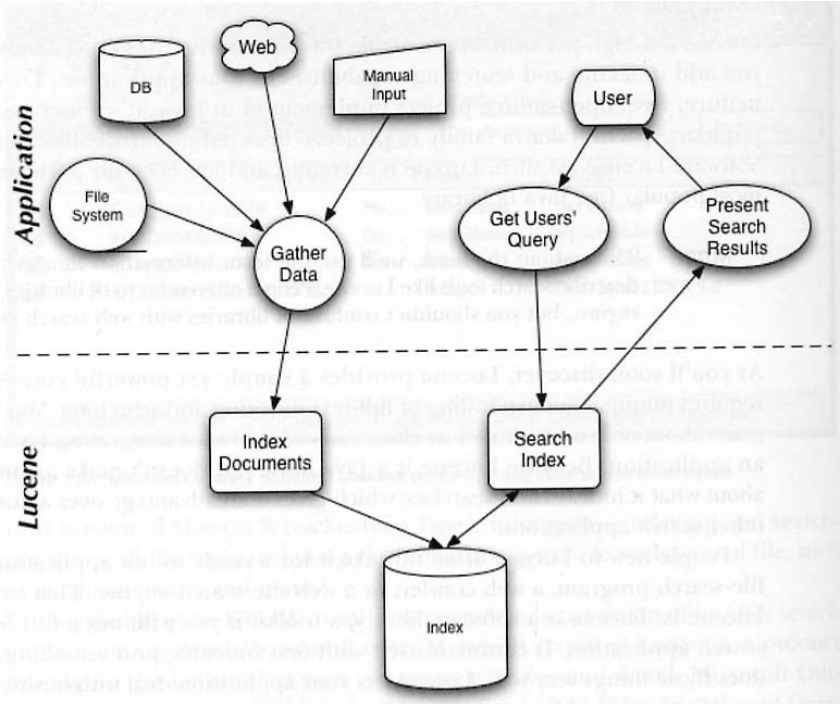


Figure 17: Communication between Lucene and applications (cf. Gospodnetić & Hatcher 2004:8).

As shown in Figure 17, there are two interfaces through which Lucene communicates with other applications, namely, the input of documents to be indexed and the parsing of queries to be used for searching the index.
The format required by the Lucene indexing framework is pure text. Therefore, every kind of document or information that can be transformed into text can be used as input for the Lucene indexing process.

Indexing takes place in three steps, wich are illustrated in Figure 18:

1. Conversion to text
2. Parsing of the documents and transformation into an internal Lucene
3. Document representation with a number of Fields (e.g. TITLE, TEXT, etc.).

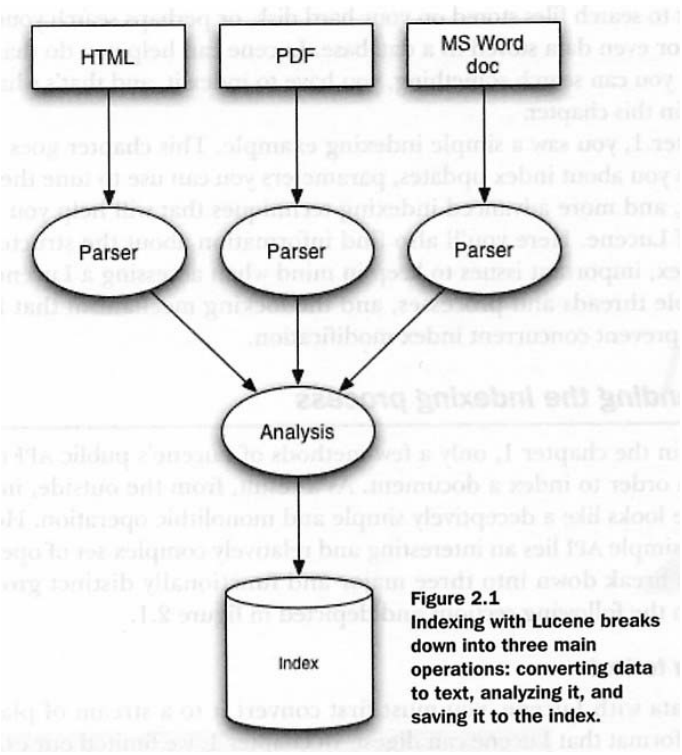


Figure 18: Indexing steps in Lucene (cf. Gospodnetić & Hatcher 2004:30).

Both documents themselves and individual document fields can be boosted (cf. similarity formula in section 4.2.3).

Analysis

"Analysis*, in Lucene, is the process of converting field text into its most fundamental indexed representation,* terms" (Gospodnetić & Hatcher 2004:103).

Before text is written to the index, it is passed through an Analyzer in order to extract tokens to be indexed. An Analyzer is the "encapsulation of the analysis process" and tokenizes text by performing any number of operations, which could include extracting words, discarding punctuation, removing accents from characters, lowercasing (normalizing), removing common words, reducing words to a root form (stemming), or changing words into their basic form (lemmatization). The analyzing process turns text into a stream of tokens, literally a Lucene TokenStream. There are two types of TokenStreams: Tokenizers and Filters. Tokenizers define which characters are to be used as separators (e.g. whitespaces for Western texts). Individual Tokens are then passed through a selection of Filters, such as lowercase filter, stop word filter, stemmers, etc.

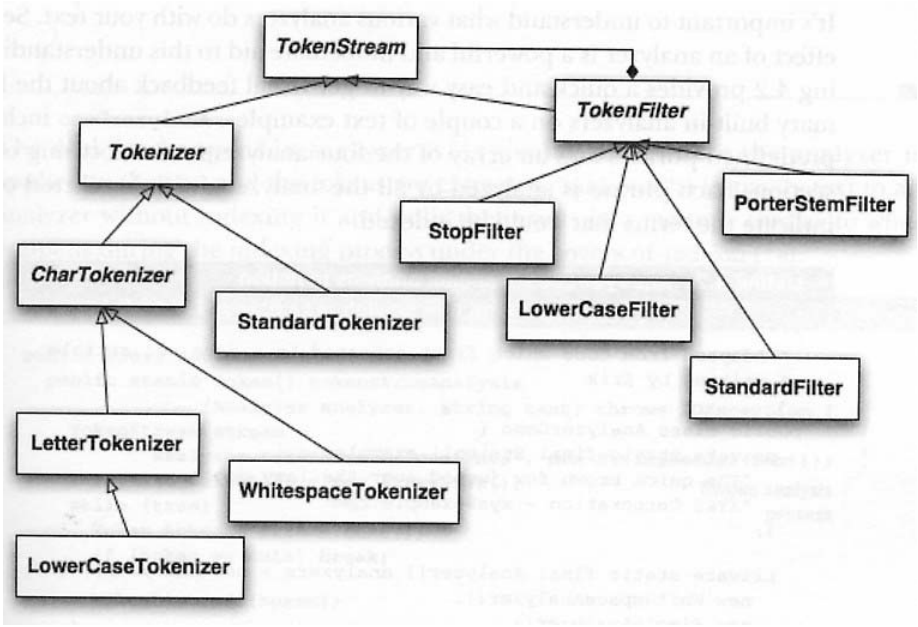


Figure 19: TokenStream class hierarchy and Analyzer building blocks (cf. Gospodnetić & Hatcher 2004:111).

A Token carries with it a text value (the word itself) as well as meta-data: the start- and end-offsets in the original text, a token type, and a position increment, as visualized in the following figure.

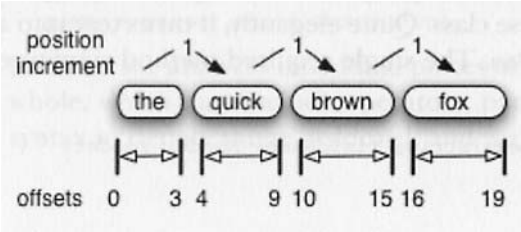


Figure 20: Positional information of a Lucene Token (cf. Gospodnetić & Hatcher 2004:108).

The token type can be used for deciding different treatment by filters. For the processing of Japanese, for example, token types KATAKANA, ROMAN and KANJI could be defined. A filter could then be created, which divides KANJI tokens into n-grams but leaves KATAKANA and ROMAN tokens as they are.

A token with a zero position increment places the token in the same position as the previous token. Analyzers that inject word aliases (cf. the WordNet option described in 5.2.4) can use a position increment of zero for the aliases. The effect is that phrase queries work regardless of which alias was used in the query.

Choosing the right analyzer is a crucial decision with Lucene, depending not only on the language(s) of the documents, but also on the text domain in question and on its

specific terminology or format (e.g. frequent use of acronyms requires special handling of groups of capital letters).

The same analysis which is performed on the documents must be applied to the queries in order to guarantee for comparability of the tokens.

Index writing

Eventually, the data is stored in an inverted index, which guarantees efficient use of disc space while allowing quick keyword lookups. Figure 21 shows the logical view of a Lucene index.

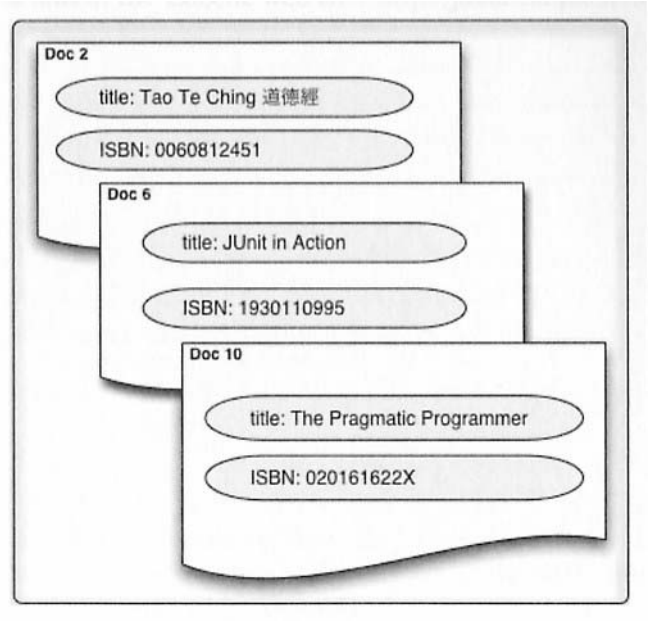


Figure 21: Logical view of a Lucene index (cf. Gospodnetić & Hatcher 2004:396).

Physically, a Lucene index consists of one or more segments, each segment being made up of several index files. This structure allows for incremental indexing. New documents are simply added to newly created index segments and only periodically merged with other, existing segments, which minimizes physical index modifications. No re-indexation of the whole corpus is necessary when new data is added. This makes Lucene suitable for large bodies of data.

Searching

Searching the index only requires query parsing for the creation of Query objects. These are then passed to an IndexSearcher object's search method. The return value is a Hits object with an ordered collection of hits (by score), providing easy and highly

efficient access to those search results. The following section introduces the different kinds of queries and search options.

4.2.2 Search Options

Lucene offers a number of query types, which are listed below:

TermQuery:

```
Term t = new Term("title", "Hildesheim");
Query query = new TermQuery(t);
```

A term is the smallest indexed piece, consisting of a field name and a text-value pair. In a TermQuery, a specified field (here: “title”) is searched for a certain keyword (here: “Hildesheim”).

RangeQuery:
Terms are ordered lexicographically within the index, allowing for efficient searching of terms within a range. RangeQuery allows searching from a starting term through an ending term. This may be useful for queries on date ranges, for example.

PrefixQuery:
Searching with a PrefixQuery matches documents containing terms beginning with a specified string. This is a helpful feature for search in categories.

BooleanQuery:
A BooleanQuery is a container of Boolean clauses, allowing AND, OR, and NOT combinations, and provides a practical way to combine queries.

PhraseQuery:
A PhraseQuery allows a maximum allowable positional distance between terms to be considered a match, e.g. maximum total number of moves allowed to put the terms in order. The scoring is based on the edit distance needed to match the phrase.

WildcardQuery:
WildcardQueries are queries with missing pieces. The operator “*” replaces >=0 characters, and the operator “?” replaces {0, 1} characters.

FuzzyQuery:
A FuzzyQuery matches terms similar to a specified term. Similarity between terms in the index and a specified target term are determined with the Levenshtein distance

algorithm[46]. The edit distance affects scoring, such that terms with less edit distance are scored higher. Equation 3 shows how the FuzzyQuery distance is calculated:

$$1 - \frac{dis\tan ce}{\min(textlen, t\arg etlen)}$$

Equation 3: Calculation of fuzzy matches (cf. Gospodnetić & Hatcher 2004:93).

The variable "targetlen" refers to the length of the target term

Since the calculation of fuzzy matches takes some time, FuzzyQuery is a feature to be employed with care.

Apart from the basic TermQuery and the BooleanQuery combination type, the FuzzyQuery seems to be a practical solution for the handling of katakana variants.

4.2.3 Similarity Calculation

Lucene is based on the vector space model and its similarity function is an extension of the TFf/IDF formula, with added boosting and normalizing factors:

$$\sum_{t\ in\ q} tf(t\ in\ d) \cdot idf(t) \cdot boost(t.field\ in\ d) \cdot lengthNorm(t.field\ in\ d)$$

Equation 4: Lucene's scoring formula (cf. Gospodnetić & Hatcher 2004:78).

The formula shows the raw score. However, scores returned from Hits objects are not necessarily the raw score. If the top-scoring document scores greater than 1.0, all scores are normalized from that score, such that all scores from Hits are guaranteed to be 1.0 or less.

Table 14 shows a description of the individual factors taken into account by the scoring formula.

[46] The Levenshtein distance is a measure of similarity between two strings, where distance is measured as the number of character deletions, insertions, or substitutions required to transform one string to another string.

Factor	Description
tf(t in d)	Term frequency factor for the term (t) in the document (d)
idf(t)	Inverse document frequency of the term
boost(t.field in d)	Field boost, as set during indexing. The boost factor affects a query's (in case of a MultipleQuery) or field's influence on the score.
lengthNorm(t.field in d)	Normalization value of a field, given the number of terms within the field. This value is computed during indexing and stored in the index
coord(q, d)	coordination factor, based on the number of query terms the document contains
queryNorm(q)	Normalization value for a query, given the sum of the squared weights of each of the query terms

Table 14: Description of the individual factors taken into account by the scoring formula (cf. Gospodnetić & Hatcher 2004).

4.2.4 Extended Features

There are a number of extended Lucene features. Some form part of the official Lucene distribution and others are accommodated in the Lucene Sandbox, a CVS repository for contributions above and beyond the Lucene core code base. This section presents the two most relevant features for adding Japanese language support, the WordNet add-on and the "more-like-this" facility based on term vectors.

The WordNet[47] add-on was recently contributed by Dave Spencer and allows the WordNet synonym database to be used as a Lucene index for rapid synonym lookup, e.g. for synonym injection during indexing or querying.

The program converts the WordNet Prolog synonym database into a standard Lucene index with an indexed field "word" and unindexed fields "syn" for each document. The size of the resulting index is approximately 2.5MB, which is compact enough to load into the RAM for quick access. For each word, synonyms can be looked up and tied into the original query (during querying) or document (during indexing). In order to ensure that all synonyms are found, normalizing and stemming steps must be employed with care.

Another interesting feature are term vectors, collections of term-frequency pairs. Term vector storage must be enabled on the desired fields during indexing. This feature makes it possible to find documents "like" a particular document, which can be used for

[47] WordNet was developed at Princeton University's Cognitive Science Laboratory, led by Psychology Professor George Miller. It illustrates the net of synonyms representing word forms that are interchangeable, both lexically and semantically.

Latent Semantic Analysis (LSI). Based on the term-frequency feature, David Spencer recently created a generic "more-like-this" facility, which is now to be found in the Lucene Sandbox. Term vectors can also be used for Query Expansion.

4.2.5 Adding Japanese Language Support to Lucene

The only language-specific step within Lucene is the analysis process, consisting of tokenizing, stemming, filtering, and stop word removal.

Lucene comes with two built-in language-specific analyzers: GermanAnalyzer and RussianAnalyzer. Additionally, there is the freely-available SnowballAnalyzer package, which supports many European languages: Danish, Dutch, English, Finnish, French, German, Italian, Norwegian, Portuguese, Russian, Spanish, and Swedish.

For the implementation of support for other languages, Lucene only provides basic building-block support, provided there is nothing to be found in the Sandbox (cf. Gospodnetić & Hatcher 2004:140). In general, the hurdles for implementing analyzers for new languages tend to be:

- character set encoding
- proper handling of reading external files
- tokenizing method
- different sets of stop words
- unique stemming algorithms
- accent removal
- language detection, if necessary

For Asian languages, the Lucene Sandbox offers a ChineseAnalyzer which splits text strings into unigrams, and a CJKAnalyzer[48] which creates bi-grams from input text. The CJKAnalyzer functions reasonably well for Japanese text, however, as Japanese uses four different kinds of scripts for very different purposes, an analyzer for Japanese should take this fact into account, e.g. by discarding hiragana characters (cf. sections 1.1.2 and 2.1.4).

As for the character set encoding problem, Lucene internally stores all characters in the standard UTF-8 encoding. It is the responsibility of the developer to read the external text into Java and Lucene. When indexing files on a file system, one must know in which encoding the files were saved in order to read them properly.

[48] CJK stands for Chinese, Japanese, Korean

4.3 MIMOR for Japanese

The following presents the approaches that were implemented for Japanese retrieval support. Since the main difference between IR with European language and Japanese is in word segmentation, the focus was set on segmentation and indexing strategies.

4.3.1 Segmentation and Indexing

In line with the "Multiple Indexing Multiple Object Representation" paradigm of MIMOR and taking into account the positive results of combination-of-evidence approaches in Japanese indexing, various indexing methods were implemented with the aim of testing their fusion. Apart from the basic n-gram-based and word-based approaches, a yomi-based indexing method was developed in order to test its effectiveness in the handling of orthographic varieties. In the following section, a short description of the implemented segmentation and indexing approaches will be provided.

N-gram-based indexing

For the creation of a bi-gram-based index, a new JNGramAnalyzer class was implemented, which differs from the CJKAnalyzer class of the Lucene sandbox in that it applies a different treatment to the individual scripts to be found in Japanese.

As in Japanese IR it has commonly been found that n=2 yields the best retrieval results (cf. section 2.1.4), the experiments were carried out with n=2.

Figure 22 shows a step-by-step illustration of the segmentation process carried out within the JNGramAnalyzer.

Key:

Green box:	katakana characters
Red box:	kanji characters
Blue box:	Roman characters
No box:	no token characters
Red font:	punctuation marks
[…]	indicating token boundaries

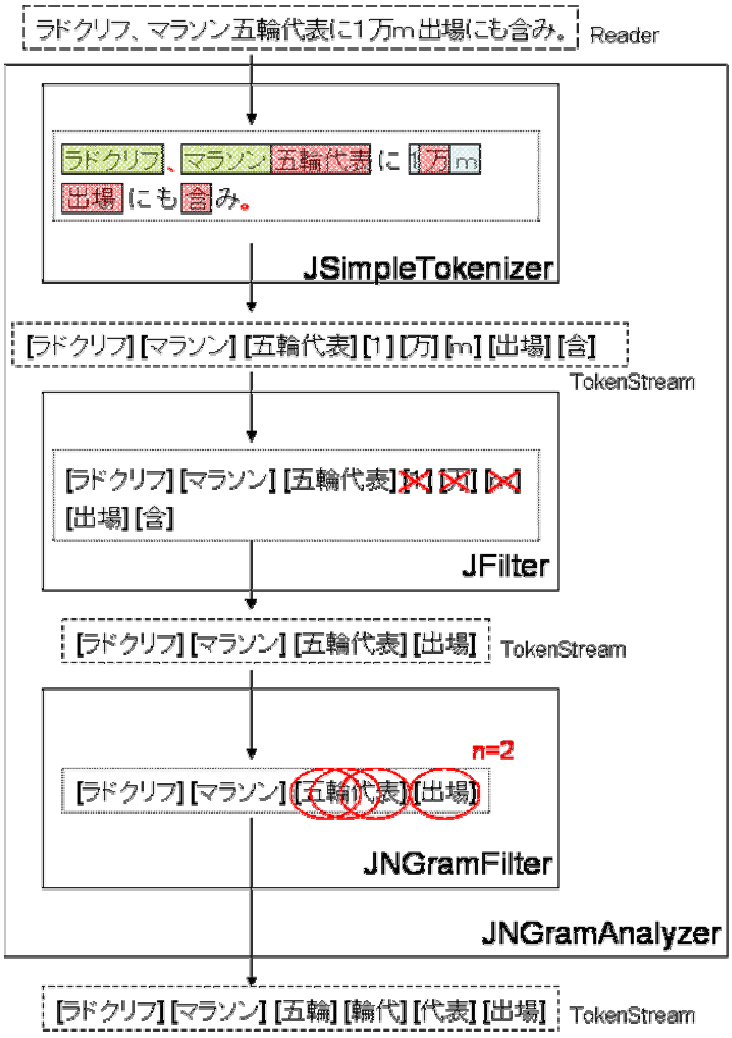


Figure 22: Bi-gram segmentation of an example sentence with the JNGramAnalyzer.

JSimpleTokenizer:

A first basic segmentation is carried out by an instance of the JSimpleTokenizer class, which divides the input string where there is a change in script and discards hiragana characters and other non-token characters.

JFilter:

The resulting TokenString is subsequently passed to a JFilter instance, where Roman character and katakana tokens are normalized (conversion to Unicode Basic Latin in the case of Roman character tokens and to Unicode Katakana Full Width in the case of katakana tokens) and one-letter katakana and Roman character tokens are filtered out.

JNGramFilter:

The next step is carried out by a JNGramFilter, which divides kanji tokens into overlapping bi-grams and leaves all other token types in their original form.

Word-based indexing

The morphological analysis for the word- and yomi-based indices is carried out with the Japanese morphological analyzer ChaSen49 (using an interface written by Michael Koch for the JGloss project50). Only nouns, verbs, and adjectives are kept as index terms. The following line had to be added to the ChaSen resource file in order to avoid a segmentation of numbers:

(COMPOSIT_POS ((名詞 数) (記号)))

Out-of-vocabulary words, i.e. words not recognized by ChaSen, re-divided into bi-grams. This can be called a hybrid approach (cf. section 1.2.2).

Yomi-based indexing

For the yomi-based index, the same morphological analysis as for the word-based index is carried out, however, not the term's surface form but its reading is kept as indexing unit.

In the case of more than one suggested reading for a term, the readings are indexed as separate terms (e.g., ナマモノ – "raw thing" and セイブツ – "living thing, LEBEWESEN" for 生物). This leads to more tokens compared to the word-based indexing method.

4.3.2 Optimization Strategies

Stoplist

A stoplist for each individual index was created determining the 100 most frequent index terms. It was decided heuristically, which of those terms should be discarded. In the case of the scientific abstracts collection, terms such as 研究 (research), 方法 (method), 実験 (experiment), 検討 (investigation, study), 結果 (result), and 目的 (purpose), were dropped. These terms act as structure words, and are to be found in practically every scientific document. Similarly, terms such as 記事 (article) and 問題 (problem) were dropped for queries within the news domain. The yomi stoplist contained some equivalents of typical stop terms that were also to be found in the word-based stop list, such as モノ (thing), as well as the numerals 0 (レイ、ゼロ) to 9 (

[49] http://chasen.aist-nara.ac.jp/hiki/ChaSen/

[50] http://jgloss.sourceforge.net/

キュウ), and a number of individual syllables. The complete stoplists can be found in Appendix A.

Pseudo Relevance Feedback

The Pseudo Relevance Feedback implementation by [Hackl 2004] was integrated into the system. Expansion terms are selected based on the Robertson Selection Value[51] (Robertson 1991), which is calculated by multiplying the term relevance (obtained with the probabilistic Robertson/Sparck Jones formula) with the number of documents containing the term (cf. Equation 5).

$$rsv(i) = r(i) \cdot rw(i)$$

Equation 5: Robertson Selection Value.

Equation 6 shows how the weight of term i is obtained using the Robertson/Sparck-Jones formula.

$$rw(i) = \log \frac{(r(i)+0.5)(N-n(i)-R+r(i)+0.5)}{(n(i)-r(i)+0.5)(R-r(i)+0.5)}$$

Equation 6: Robertson/Sparck-Jones Formula (Robertson & Sparck-Jones 1976).

R = the total number of relevant documents
N = the total number of documents
r(i) = the number of relevant documents containing term i
n(i) = the total number of documents containing term i

As in the CLEF2004 implementation, the number of relevant documents to be retrieved and the number of terms to be extracted can be specified in the searching.properties file.

Fuzzy Querying

In an attempt to achieve a flexible handling of katakana variants (cf. sections 2.2.3 and 3.3.4), a Fuzzy Querying option was implemented using Lucene FuzzyQuery. Fuzzy Querying is only employed with the word-based index and is exclusively applied to

[51] Unfortunately, the Robertson Selection Value shares its abbreviation "RSV" with the Retrieval Status Value

katakana terms, that is, during query generation, the script type of a search term is determined and a “FuzzyQuery” is created, if a katakana term is found.

4.3.3 Fusion Approaches

In order to achieve an effective combination of the individual indices described in section 4.3.1, three different fusion algorithms were implemented:

1. Raw Score:
   The results are ordered exclusively by numerical value. Consequently, it is possible that more result documents are chosen from one result list than from another, if the first yields higher result values.

2. SumRSV:
   The score of a result document is calculated as the sum of its retrieval status value (RSV) multiplied by the weight attached to the system. SumRSV = ∑ αI · RSVi, where αI may be used to represent the weight of an index.

3. Z-score:
   Z-score fusion allows for a normalized linear combination of the search results (cf. Savoy 2004). The contribution of each individual systems is controlled using a weight represented by the parameter α (see. Equation 5).

$$Z-ScoreRSV_k = \alpha \cdot \left[\frac{RSV_k - Mean^i}{Stdev^i} + \delta^i \right]$$

$$\delta^i = \frac{Mean^i - Stdev^i}{Stdev^i}$$

Equation 7: Z-score.

Key: RSV stands for Retrieval Status Value, the score assigned to a retrieved document

Raw Score can be considered as the most basic fusion strategy and was mainly used for first testing. It also outperformed RoundRobin fusion in experiments carried out in the CLEF context by a project group at the University of Hildesheim for the fusion of the result lists of the individual languages. SumRSV and Z-Score were successfully employed by [Savoy 2004] in NTCIR-4, Z-Score yielding especially promising results.

The contribution of the individual indices to the final result set is controlled by weights, which can be specified in the searching.properties file. The fusion approach to be applied can be specified as an argument of the Searcher.java class, where the search process is launched.

Figure 23 illustrates the process from the generation of a query to its distribution and finally the fusion of the result lists returned by the different indices.

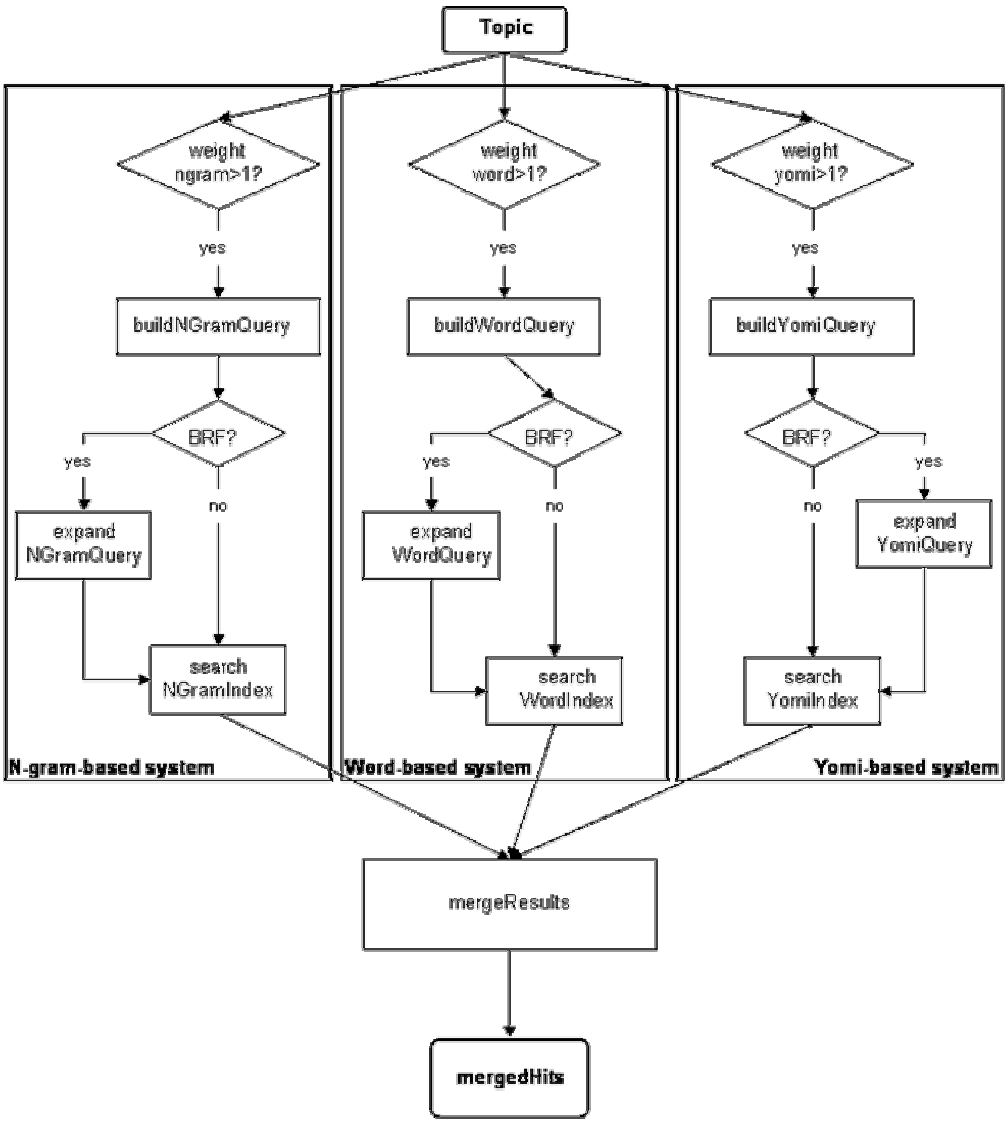


Figure 23: Flow chart of fusion process.

4.3.4 Translation

A translation module was implemented, which provides the basic functionality for Japanese/English query translation.

Figures 24 and 25 illustrate the query translation process including pre- and post-translation options. In the case of English-Japanese CLIR, the Japanese query needs to be distributed to the different indices for searching.

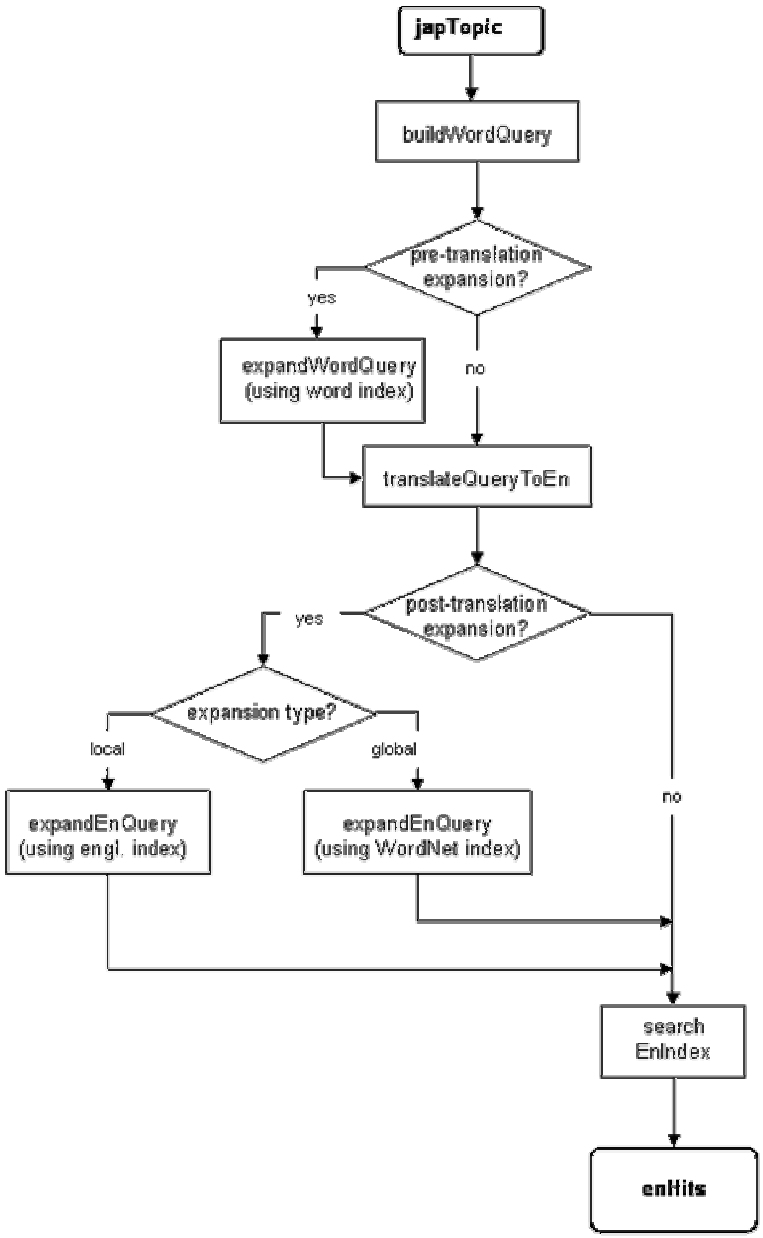


Figure 24: Cross-lingual Japanese-English translation and querying.

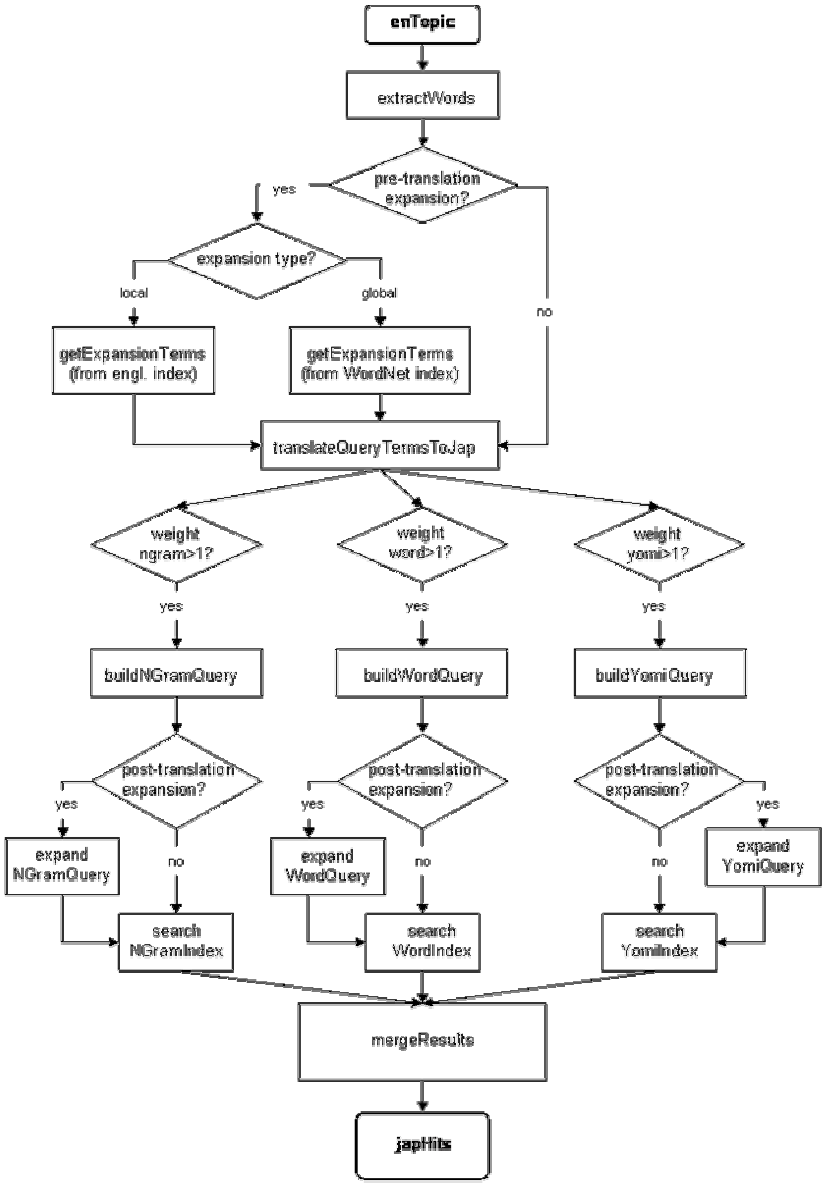


Figure 25: Cross-lingual English-Japanese translation and querying.

Translation is carried out using the freely available EDICT dictionary and its ENAMDICT supplement (cf. section 3.3.1), and also an interface class from jgloss by Michael Koch, which effects the lookup of words in the EDICT text file and provides a list of translation terms. With the WaDokuJT being supplied in the same format, Japanese/German translation could be effected with the same module with only minor adaptations. The global PRF functionality uses the WordNet extension of Lucene in order to obtain synonyms of the English search terms.

So far, the query translation module has only been evaluated formatively. In a next step, further experiments are necessary in order to determine how accurately translation functions, what percentage of terms is found in the dictionary, in which combination and with which parameters pre- and post-translation work best, and which further strategies could be employed in order to handle OOV terms.

5 Experiments and Analysis

The following chapter covers the experiments carried out with the system and provides an analysis of the results.

5.1 The NTCIR Test Collection

5.1.1 Collections Used for Testing

All experiments were carried out using sub-collections of the NTCIR test collection. In order to compare retrieval results across different document domains, the corpus of scientific abstracts used in NTCIR-1 (hereafter called “NTCIR-1”) and the collection of newspaper articles extracted from Mainichi Shimbun 1998 (hereafter called “Mainichi’98”), a part of the test collection used in NTCIR-4, were chosen as examples of different text genres with different characteristics.

Major newspaper companies in Japan have strict guidelines concerning vocabulary and orthography. Consequently, the Mainichi’98 corpus should be an example of a standardized and homogeneous document collection.

Scientific articles, on the other hand, are written by a great number of authors with various writing styles, variable wording, and different orthography. In the scientific domain, new words are continuously coined, often by creatively combining new compound words or by transcribing English words into Japanese (using the katakana script). However, the usage of these new words is not always consistent. Consequently, a greater number of compound words and OOV terms as well as more orthographic variation can be expected in the NTCIR-1 collection.

These characteristics might result in differences in the retrieval performance of the individual approaches tested. Strategies that account for orthographic varieties should have more effect on the NTCIR-1 collection. Similarly, the n-gram indexing approach should work better for the scientific abstracts, as these might contain a number of words that are not recognized by the morphological analysis tool.

The NTCIR-1 and Mainichi’98 collections contain 332,918 and 115,552 documents, respectively.

5.1.2 Structure of NTCIR Topics and Documents

Table 15 shows the tags used for identifying each NTCIR-4 document. For indexing purposes, only the document identifier (<DOCNO>), the title (<HEADLINE>), and the text (<TEXT>) fields are of importance.

Mandatory Tags	
<DOC>	The tag for each document
<DOCNO>	Document identifier
<LANG>	Language code: CH, EN, JA, KR
<HEADLINE>	Title of this news article
<DATE>	Issue date
<TEXT>	Text of news article
Optional Tags	
<P>	Paragraph marker
<SECTION>	Section identifier in original newspapers
<AE>	Contain figures or not
<WORDS>	Number of words in 2 bytes (for Mainichi Newspaper)

Table 15: Tags used for identifying document fields (Kishida et al. 2004).

NTCIR-1 documents have a slightly different SGML structure. In this case the title, abstract, and keyword fields are used for indexing and searching.

Table 16 explains the tags used for identifying the topic fields. The <TITLE>, <DESC>, <NARR>, and <CONC> fields can be used for automatic query generation. In the following experiments, queries were constructed from all four fields.

Tag	**Description**
<TOPIC>	The tag for each topic
<NUM>	Topic identifier
<TLANG>	Source language code: CH, EN, JA, KR
<TLANG>	Target language code: CH, EN, JA, KR
<TITLE>	The concise representation of information request, which is composed of noun or noun phrase (ibid.)
<DESC>	A short description of the topic. The brief description of information need, which is composed of one or two sentences (ibid.)
<NARR>	A much longer description of topic. The <NARR> may has three parts; (1) <BACK>…</BACK>: background information about the topic is described (2) <REL>…<REL>: further interpretation of the request and proper nouns, the list of relevant or irrelevant items, the specific requirements or limitations of relevance, and so son are given. (3) <TERM>…</TERM>: definition or explication of proper nouns, scientific terms and so on (ibid.)
<CONC>	The keywords relevant to whole topic (ibid.)

Table 16: Tags used for identifying topic fields (Kishida et al. 2004).

Figure 26 shows an example topic.

```
<TOPIC>
<NUM>003</NUM>
<SLANG>CH</SLANG>
<TLANG>JA</TLANG>
<TITLE>ES細胞</TITLE>
<DESC>ヒトES細胞の紹介記事を探したい</DESC>
<NARR>
<BACK>2つのアメリカの研究グループが，実験室でヒトES細胞の培養に成功した。かれらはこの研究が心筋から脳組織まで，いかなる組織の培養にも応用できると考えている。医学におけるES細胞の用途と特性について，また，倫理論争があるのかどうか，論争があるのであればどのようなものなのかについて知りたい。</BACK>
<REL>用途，医学的特性，倫理論争の紹介を含む文書を適合とする。科学者の研究過程は適合しない。</REL>
</NARR>
<CONC>ES細胞，医学的特性，用途，倫理，論争</CONC>
</TOPIC>
```

Figure 26: Topic No.3 (NTCIR-4).

5.1.3 Relevance Judgments in NTCIR

In contrast to TREC and CLEF, the NTCIR workshop does not adopt binary relevance judgment. Instead, each document is assigned one out of four levels of levels of relevance in the judgment process:

S: “highly relevant”
A: “relevant”
B: “partially relevant”
C: “irrelevant”

This is done in order to guarantee a higher level of measurement granularity. Since the trec_eval[52] program is used to generate the evaluation results and this program adopts binary relevance, these four levels need to be transformed into a binary judgment. Therefore, two relevance judgment files are provided: “rigid” and “relaxed”. The former lists only “highly relevant” and “relevant” documents (S+A), while the latter also lists “partially relevant” documents (S+A+B).

The evaluation of the following experiments is based on the relaxed relevance judgments.

[52] trec_eval is the program used in the Text REtrieval Conferences to generate the evaluation results

5.1.4 Adaptations

The complete Japanese NTCIR-4 collection contains years 1998 and 1999 of Mainichi Shimbun and Yomiuri Shimbun. Therefore, for some queries there were not sufficient, i.e. less than five, relevant documents contained in the sub-corpus Mainichi'98. These topics were identified and removed. The remaining 46 topics were used for the experiments.

5.2 Evaluation of Basic Indexing Strategies

5.2.1 Overview

Table 17 to 19 contain an overview of the computation time necessary for the creation of the individual indices, of the storage space needed, and of the number of terms per index. All calculations were carried out on a 3GHz Intel Pentium 4 with 512MB RAM.

	Mainichi'98	NTCIR-1
N-gram	<2h	<2h
Word	~17.5h	~8.3h
Yomi	~17.5h	~7.7h

Table 17: Calculation time per index.

The calculation time per index is only approximate, as other processes were running simultaneously on the same machine. Nevertheless, the numbers give an impression of the dimensions.

It is evident that n-gram indexing is considerably faster than word- and yomi-based indexing, which both involve the process of morphological analysis.

Word- and yomi-based indexing are comparable in speed. Objectively, yomi-based indexing should be slightly slower than word-based indexing, as both use the output of the same morphological analysis process, but for the yomi-based index sometimes several readings have to be extracted per word.

	Mainichi '98 (146 MB)	NTCIR-1 (311 MB)
Word	356 MB	628 MB
Yomi	390 MB	706 MB
N-gram	355 MB	649 MB

Table 18: Index sizes.

Surprisingly, the n-gram index is not the largest one. Although both yomi- and word-based index are produced using the same morphological analysis output, they differ in size. This can be explained by the fact that in case of several suggested readings per word, all are used as index terms. This leads to more single tokens compared to the word-based index (cf. Table 19).

	Tokens	Types	Tokens/Type
Word	21,426,876	94,124	227.6
Yomi	23,413,585	70,680	331.3

Table 19: Type-Token Ratio of the word-based and yomi-based indices (obtained from older version of indices).

The low number of yomi types compared to the number of word types reflects the abundance of homophones in the Japanese language.

5.2.2 Performance Using the Mainichi'98 Collection

Table 20 shows the average precision reached by each index. The detailed list of results per topic can be found in Appendix B, Table B1.

Index Type	N-gram	Word	Yomi
MAP	.3822	.3638	.3704

Table 20: MAP per index type (Mainichi'98).

Considering only the average precision, the n-gram index performs best, however not significantly better than the word- and the yomi-based index (T-test, confidence level 95%). The recall-precision graph in Figure 27 reveals that in fact both yomi- and word-based index outperform the n-gram index for recall levels 0.0 to 0.2, the yomi-based index yielding the highest precision values (cf. Appendix B, Table B2 for 11-point Precision values).

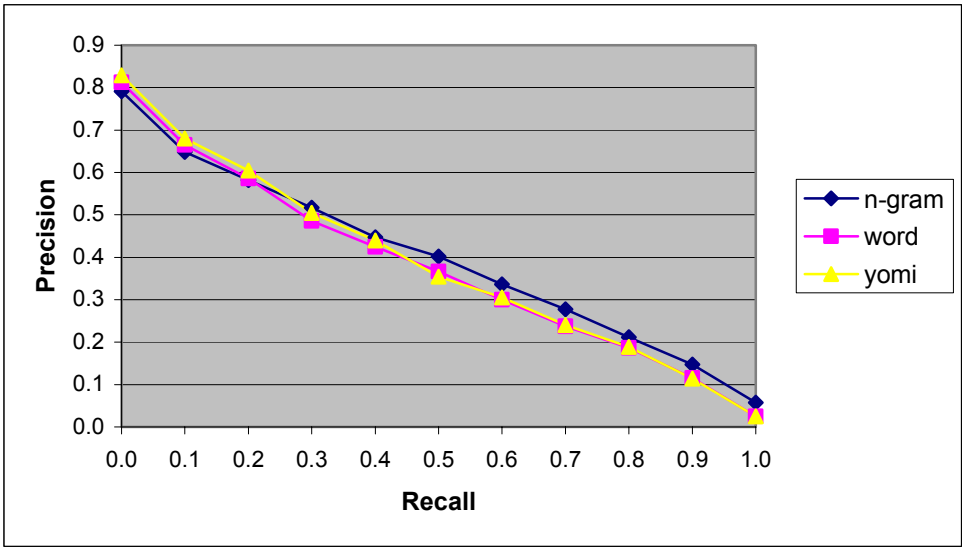


Figure 27: 11-point Precision Graph of individual indices (Mainichi'98).

This is even more salient in Figure 28. With the yomi-based index, more relevant documents are returned in the first positions of the result list. This is a desirable feature, as the typical user will not browse more than a few documents in the list. This feature is also advantageous for BRF techniques, were the first N documents are assumed to be relevant and used to extract expansion terms.

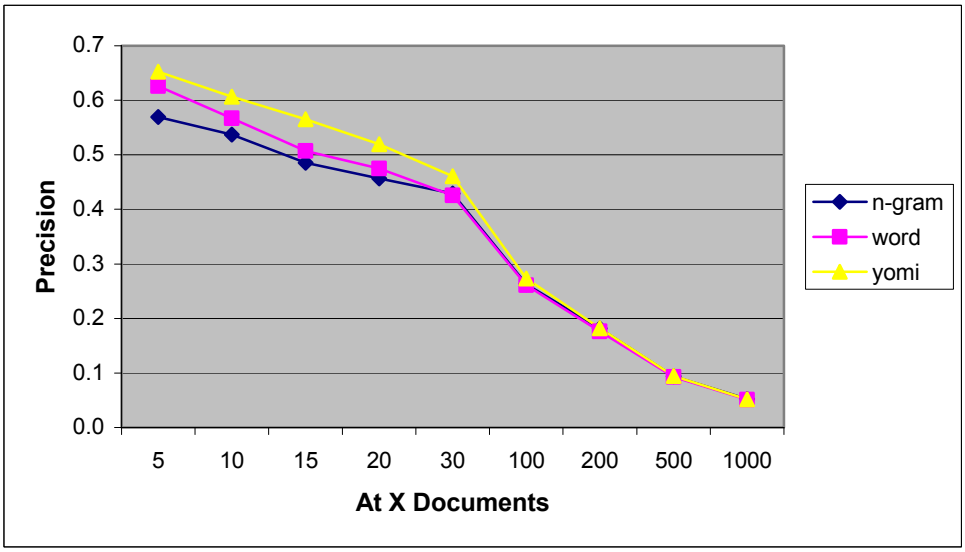


Figure 28: Frozen Ranks Graph of individual indices (Mainichi'98).

5.2.3 Performance Using the NTCIR-1 Collection

Compared to the experiments with the Mainichi'98 collection, the performance of all three indices is considerably lower for the NTCIR-1 collection (cf. Table 21). The

marked difference in retrieval performance across the two collections probably owes to the fact that the system had originally been designed to handle newspaper articles. Nevertheless, the ranking of the systems according to their MAP value is the same: the n-gram index yields the best average precision, and the yomi-based index slightly outperforms the word-based index.

Index Type	N-gram	Word	Yomi
MAP	.2971	.2622	.2722

Table 21: MAP per index type (NTICR-1).

In contrast to the experiences with the Mainichi'98 collection, however, both 11-point Precision Graph (cf. Figure 29, Table B4 in Appendix B) and Frozen Ranks Graph (cf. Figure 30) show the n-gram based index yielding the best results. Differences between the index performances are again not statistically significant (T-test, confidence level 95%), however.

The average precision values per topic can be found in Appendix B, Table B3.

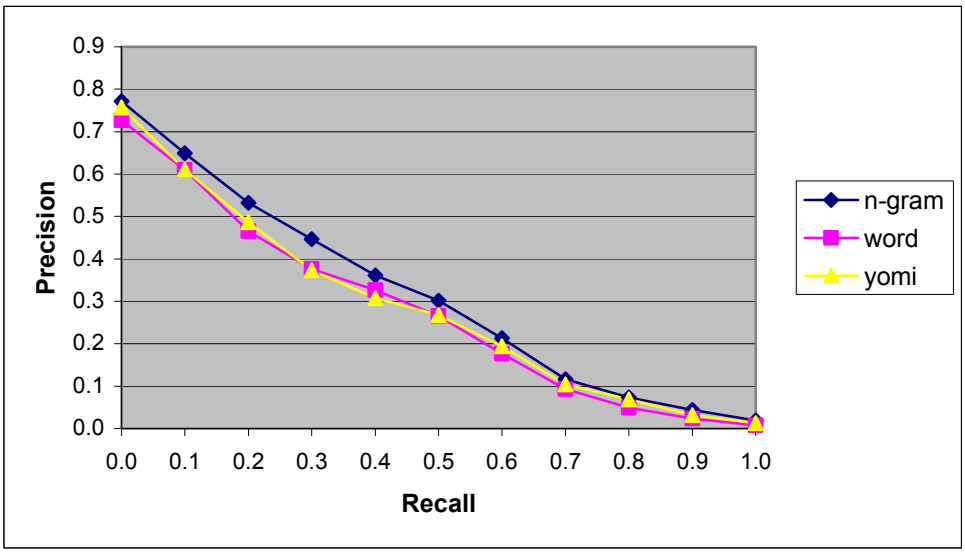


Figure 29: 11-point Precision Graph of individual indices (NTCIR-1).

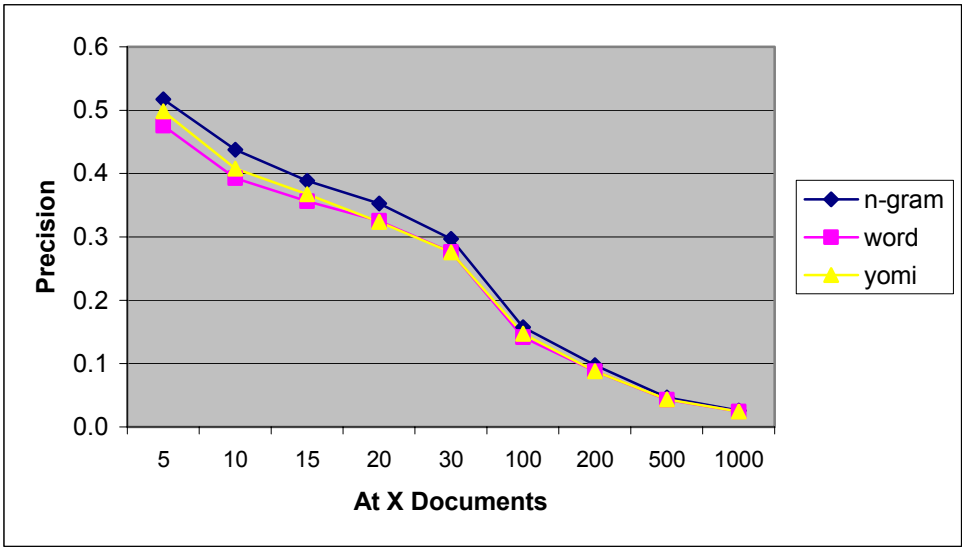


Figure 30: Frozen Ranks Graph of individual indices (NTCIR-1).

5.2.4 Analysis

The n-gram-based index yielded the highest MAP for both document collections and consistently outperformed the yomi- and word-based indices on all recall levels in the experiments with the NTCIR-1 collection.

With the Mainichi'98 collection, however, both yomi- and word-based index achieved higher average precision values for lower recall levels.

The advantage of the n-gram based segmentation and indexing approach is its independence from (the quality of) linguistic resources. Both word- and yomi-based system heavily depend on the performance of the morphological analyzer. With the Mainichi newspaper article collection being one of the corpora used for training of ChaSen, it is to be expected that ChaSen works more reliably with Mainichi'98 than with NTCIR-1. Moreover, the NTCIR-1 collection contains more new words and uncommon compounds, which are difficult for ChaSen to handle.

Therefore, it can be concluded that the non-linguistic n-gram approach generally shows the best performance, but that yomi- and word-based systems perform comparably well or better for low recall values, provided that the morphological analysis is carried out correctly.

As both depend on the output of the morphological analysis, word- and yomi-based indices generally show a similar performance. Although not significantly, the yomi-based index proved slightly superior to the word-based index with regard to MAP, 11-point Precision, and Frozen Ranks. This is a surprising result, as the high number of homophones in Japanese was rather expected to lead to a loss in precision. Further

analyses are needed to determine what accounts for the advantages of the yomi-based over the word-based system.

5.3 Optimization Experiments

5.3.1 PRF Experiments with Mainichi'98

PRF was tested with different parameters for D (number of documents used for expansion) and T (number of expansion terms). The values for D and T were gradually altered in order to determine the optimal combination.

Whereas almost no improvement could be reached for the n-gram- and word-based indices, PRF greatly boosted the MAP value for the yomi-based index. The best result was reached with D=10 and T=100, as shown in Table 22.

A possible reason for the good performance of PRF with the yomi-based index might be the high precision reached by the yomi-based index for the first documents returned (cf. Figure 28). This increases the probability of extracting relevant terms for query expansion.

	N-gram	Word	Yomi
Basic	**.3822**	**.3638**	**.3704**
+D5T30	.3986*	.4077*	.4153*
+D10T30	.4039*	.4066*	.4308*
+D10T20	.4032*	.4000*	.4240*
+D10T40	.4005	.4095*	.4359*
+D10T50	.3995	.4094*	.4379*
+D10T60	---	---	.4389*
+D15T50	---	---	.4373*
+D10T70	---	---	.4393*
+D10T100	---	---	.4407*
+D10T100	---	---	.4399*

Table 22: MAP values for different PRF parameters (Mainichi'98).

The values marked with an asterisk are statistically significant (T-test, confidence level 95%). The detailed lists of average precision values reached per topic can be found in Appendix B, Tables PRF1-PRF3.

5.3.2 PRF Experiments with NTCIR-1

As Table 23 shows, PRF did not prove very effective with the NTCIR-1 collection. For the word-based index, no improvement at all could be reached, for the n-gram and yomi-based index, improvement was only minor and not statistically significant (T-test, confidence level 95%).

Possible reasons for the low performance of PRF might be the low percentage of relevant documents returned in the first ranks, i.e. about .5 and .4 for the first 5 and 10 documents, respectively (cf. Figure 30). Furthermore, an NTCIR-1 document is typically very short, containing only about 50-60 different terms (types), compared to about 150 for a Mainichi'98 article. Therefore, the maximum number of expansion terms per document is quickly reached (cf. the identical results for +D5T30 and +D5T50).

	N-gram	Word	Yomi
Basic	**.2971**	**.2622**	**.2722**
+D10T30	.3017	.2537	.2475
+D5T30	.3079	.2558	.2684
+D5T50	.3079	.2558	.2684
+D3T30	.2998	.2594	.2739
+D20T30	.3004	.2548	.2508
+D10T10	.3017	.2537	.2473

Table 23: MAP values for different PRF parameters (NTCIR-1).

Tables PRF4-PRF6 in Appendix B contain the detailed lists of average precision reached per topic.

5.3.3 Fuzzy Querying

As explained in section 1.2.7, Fuzzy Querying might be effective in the handling of katakana variants, which represent slightly differing transcriptions of foreign words. The following experiments were carried out with the word-based index, effecting a (Lucene) FuzzyQuery every time a katakana word is used as a search term and generating normal Queries from all other search terms.

Since the NTCIR-1 corpus is more heterogeneous than the Mainichi'98 corpus and also contains more technical terms, which are often English loan words, it can be expected that the Fuzzy Querying approach will work better with the NTCIR-1 corpus.

FuzzyQuerying did not result in any improvement of the MAP, neither for the NTCIR-1 collection of scientific abstracts, nor for the Mainichi'98 newspaper corpus. Table 24 contains the MAP values of Fuzzy Querying compared to the basic word-based system for both collections.

	Mainichi'98		NTCIR-1	
	Word	**Word+fuzzy**	**Word**	**Word+fuzzy**
MAP	.3638	.3577	.2622	.2235

Table 24: MAP values for FuzzyQuerying compared to basic word-based querying.

Figures 31 and 32 illustrate the topic-per-topic performance of the basic word-based index and the basic word-based index with Fuzzy Querying (see Appendix B, Tables FQ1 and FQ2 for the average precision reached per topic).

For the tests with the Mainichi'98 collection, a slightly negative influence of Fuzzy Querying can be observed. In only two cases does Fuzzy Querying clearly outperform the basic system. An analysis of the NTCIR-4 Topic 54 used with the Mainichi'98 collection[53] revealed that one of the katakana query terms was ファイバ (faiba = fiber). However, the index contained only its variant ファイバー (faibaa). The variant was contained in 407 abstracts, 96 titles, and 203 keyword fields, while not a single document contained the original search term.

In the experiments with the NTCIR-1 collection, the effect of Fuzzy Querying was generally negative.

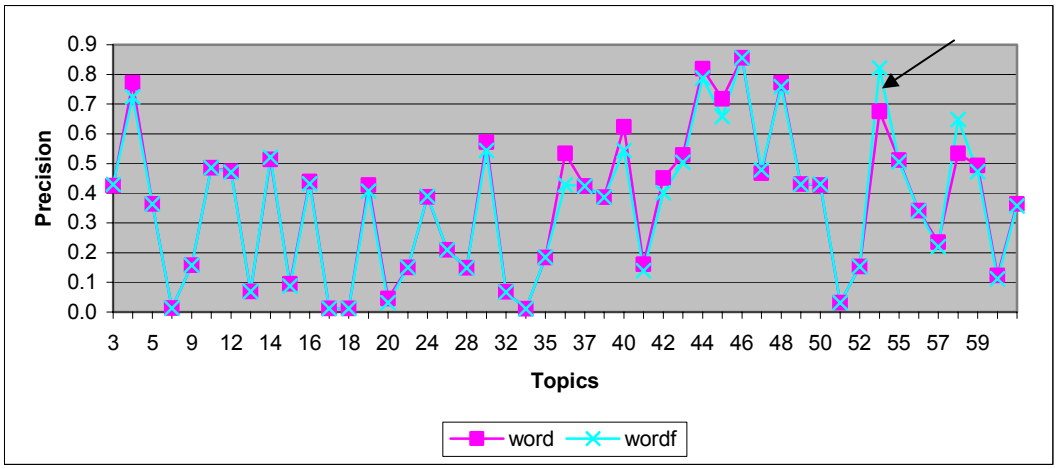


Figure 31: Topic-per-topic performance of Fuzzy Querying compared to the basic word-based system (Mainichi'98).

[53] Marked with an arrow in Figure 31

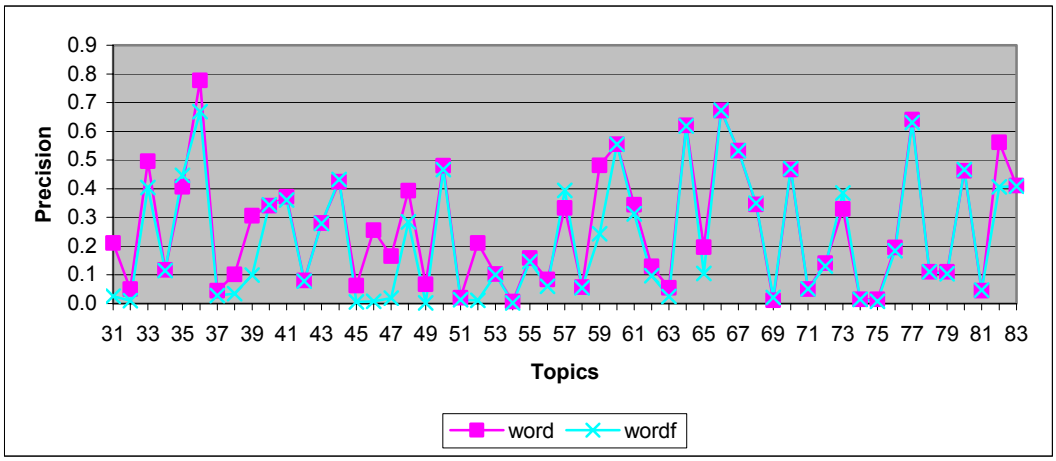


Figure 32: Topic-per-topic performance of Fuzzy Querying compared to the basic word-based system (NTCIR-1).

Since the Fuzzy Querying strategy neither led to an increase of MAP nor greatly outperformed the basic word-based system in a sufficiently high number of single cases, this approach was not further investigated.

5.4 Fusion Experiments

5.4.1 Experimental Setup

The results of the experiments described in 5.2 and 5.3 were taken into account when setting up the fusion experiments.

Since PRF proved very effective for the Mainichi'98 collection, PRF was carried out with the best-performing values per index type, that is, D=10 and T=30 for the n-gram-based index, D=10 and T=40 for the word-based index, and D=10 and T=100 for the yomi-based index.

For the experiments with the NTCIR-1 collection, no PRF was applied, as it had not lead to any significant improvement in the preceding experiments.

The first fusion run was carried out assigning the same basic weight of unity to each of the indices. Subsequently, the weights were tuned manually in order to optimize the fusion result, using the heuristic that indices which performed better in the single runs should be assigned a higher weight. This strategy imitates the MIMOR learning approach and should be automated at some point.

The fusion strategy adopted was Z-Score, which was successfully employed by [Savoy 2004] in the NTCIR-4 workshop and yielded the best results in an earlier study [Kummer et al. 2005].

The results of the best single runs, i.e. the yomi-based system applying PRF with a MAP value of.4407 for the Mainichi'98 collection and the n-gram-based system with a MAP value of .2971 for the NTCIR-1 collection, were used as a baseline for comparison.

Comparing the fusion runs with the best single run helps to determine how much increase in retrieval performance can be achieved with fusion at the cost of having several indices and requiring more time for query processing. Whereas the single-index runs take less than 10 minutes, a fusion run takes about 6 times as long.

5.4.2 Results and Analysis

Tables 25 and 26 show the MAP reached by the fusion runs for the Mainichi'98 and the NTCIR-1 collection, respectively. The detailed lists of average precision reached per topic can be found in Appendix B, Tables F1 and F2.

Weights			Avg. Precision
N-gram+PRF	Word+PRF	Yomi+PRF	
1	1	1	.4542
2	1	3	.4579
2	0	3	.4584
1	0	1	.4554
Single yomi-based system+PRF:			.4407

Table 25: MAP of the fusion runs (Mainichi '98 collection).

Fusion slightly increases the MAP value for the runs with the Mainichi'98 collection. However, the improvement over the performance of the best single system is not statistically significant (T-test, confidence level 95%). Interestingly, the best result is yielded by a combination of exclusively the n-gram- and yomi-based systems.

Weights			Avg. Precision
N-gram	Word	Yomi	
1	1	1	.3107
3	1	1	.3121
2	1	1	.3141*
3	1	2	.3127
1	0	1	.3092
Single n-gram-based system:			.2971

Table 26: MAP of the fusion runs (NTCIR-1 collection).

In the experiments with the NTCIR-1 collection, a small improvement in average precision could be reached. The performance of the combination marked with an asterisk is significantly better than the single n-gram-based system (T-test, confidence level 95%).

Figures 33 and 34 visualize the topic-per-topic performance of the single runs contributing to the fusion result, plotted against the performance of the best fusion runs, thus illustrating the cases in which fusion outperforms single runs.

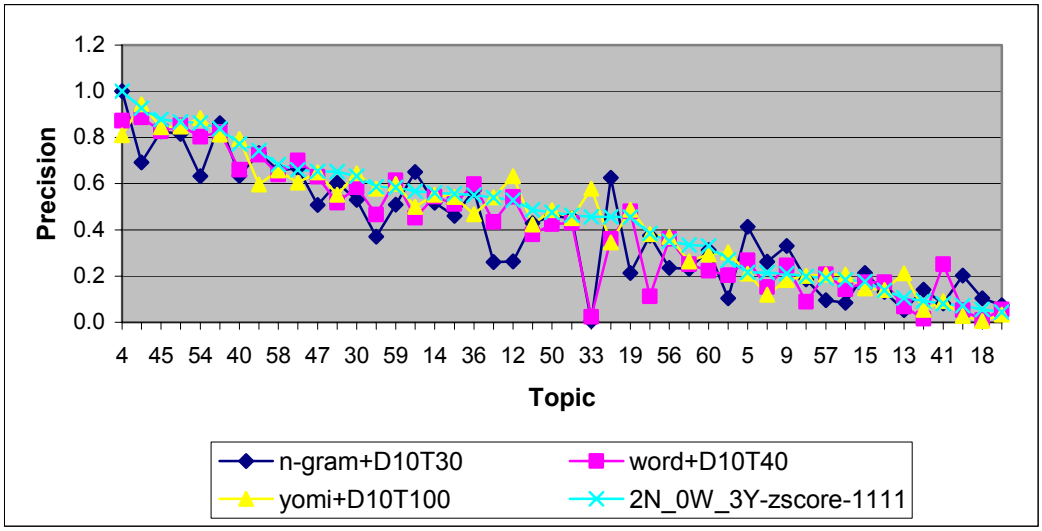


Figure 33: Average precision of the single runs used for fusion (Mainichi'98).

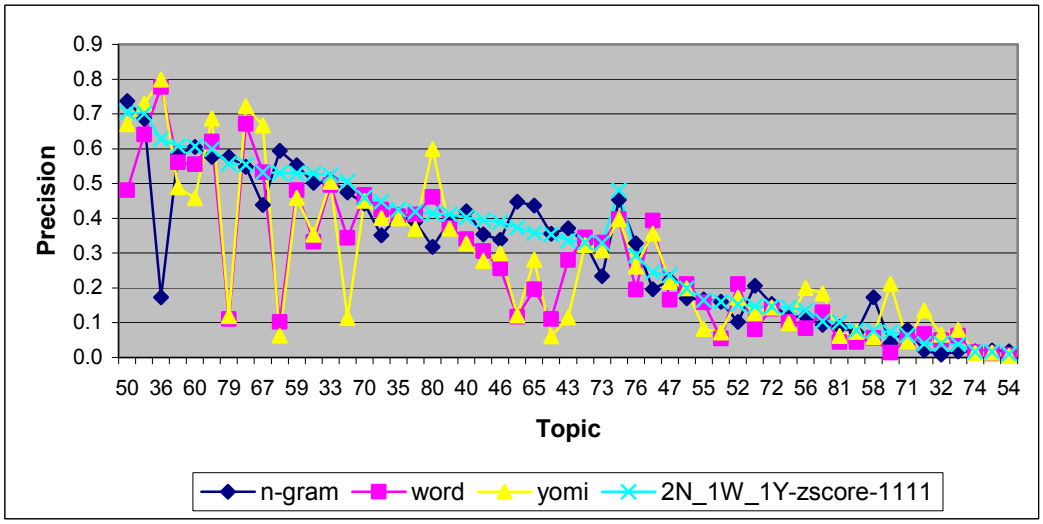


Figure 34: Average precision of the single runs used for fusion (NTCIR-1).

It can be seen that the differences between the individual indices are more marked for the runs with the NTCIR-1 collection and that the fusion run tends to be closer to the respective best-performing single run, but that there are still a number of cases where

the best-performing single approach clearly outperforms the fusion approach. This situation suggests that it would be desirable to know beforehand which approach will perform well with a certain query.

Fusion improved precision slightly with the Mainichi'98 collection and even significantly with NTCIR-1.

Interestingly, the best fusion run with the Mainichi'98 collection was the one coupling only the n-gram- and yomi-based systems. Having one less system to process the query improves retrieval speed.

6 Conclusion and Outlook

Japanese language support could successfully be integrated into the MIMOR framework. With text segmentation representing a particular challenge in Japanese IR, special attention was paid to the relative performance of individual segmentation and indexing approaches.

Research in Japanese IR has time and again proved that it is not possible to decide whether word-based methods, which require complex NLP techniques, or simple, language-independent n-gram approaches are superior. Instead, their performance varies case-by-case, depending on collection- or query-specific characteristics. In particular, the amount of non-covered vocabulary plays an important role in the performance of word-based methods. N-gram approaches, in contrast, are very robust, but neglect word-level semantics.

This is an interesting situation for MIMOR with its “multiple-indexing-multiple-object-relation” approach, which aims at identifying and combining the best-performing technique(s) to retrieve document objects.

Therefore, for the integration of Japanese language support it was decided to implement three different indexing strategies (i.e. bi-gram-, word-, and yomi-based indexing), and to investigate their individual performance as well as the benefits of their combination.

Experiments were carried out with two different document genres, the NTCIR-1 collection of scientific abstracts as an example of a rather heterogeneous corpus with many unknown words, and the Mainichi Shimbun articles of the year 1998 as an example of a rather standardized corpus which can be handled well by a morphological analyzer.

The results confirmed that there is no single best approach, but rather that the performance depends on query and collection features.

When comparing the basic approaches, bi-gram-based indexing showed the best performance for both collections. However, it could be observed that both yomi- and word-based systems yielded better precision values for the first documents retrieved in the Mainichi’98 collection.

This phenomenon had an interesting effect in the PRF experiments: with the Mainichi'98 collection, the yomi-based system in particular experienced a great boost in precision through query expansion.

With the NTCIR-1 corpus, however, the comparably weak performance at low recall levels, along with the short length of the documents, led to a poor performance of PRF.

In the fusion experiments, significant improvement could only be reached for the NTCIR-1 corpus. The results for the Mainichi'98 corpus showed an interesting phenomenon: the best-performing run was a combination of exclusively n-gram- and yomi-based indices. As searching in several indices and the fusion of result lists take quite some time, it is advantageous to know if a system does not contribute sufficiently to the retrieval performance.

The results clearly showed that there are differences in system performance with respect to document collection characteristics. In particular, the degree of standardization of vocabulary seems to play a vital role for the performance of linguistic approaches like word- and yomi-based indexing. These achieve rather high precision values for the first documents retrieved with the Mainichi'98 newspaper collection. The n-gram approach, on the other hand, proved more robust with the NTCIR-1 collection of scientific abstracts, which contain a high number of OOV terms.

It can be concluded that the optimal combination of systems, approaches, and/or parameters depends on the individual case. It is therefore desirable to be able to select the most "productive" strategy in a flexible way, based on an analysis of the problem type (e.g. length of query, characteristics of document collection, etc.).

The most productive strategy might be one single approach with specific parameters (e.g. number of documents and terms used for PRF), a combination of two or more approaches with the appropriate weights, or the re-organization of system components (e.g. fusion on several stages or selection of documents used for query expansion with an approach yielding high precision values).

At this point, a detailed case-by-case analysis of topics and results is necessary in order to explore which characteristics of a document or query account for the positive or negative performance of a certain strategy. Secondly, it needs to be determined how these characteristics could be identified automatically.

Apart from a detailed analysis, it would be desirable to reproduce the experiments with a larger test collection such as the complete NTCIR-4 corpus or the current NTCIR-5 collection in order to assess the system in relation to the performance of other systems.

Works Cited

[Ballesteros & Croft 1997]
Lisa Ballesteros & Bruce Croft (1997): *Phrasal translation and query expansion techniques for cross-language information retrieval*. In: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States, pp. 84 – 91.

[Ballesteros & Croft 1998]
Lisa Ballesteros & Bruce Croft (1998*): Resolving ambiguity for cross-language retrieval*. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, pp. 64 – 71.

[Breen 2004]
James Breen (2004): JMdict: a Japanese-Multilingual Dictionary. In: Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources, Geneva, Switzerland, August 28, 2004.

[Buckley et al. 1998]
Chris Buckley, Mandar Mitra, Janet Walz & Claire Cardie (1998): *Using clustering and superconcepts within SMART*: TREC 6. In: The Sixth Text REtrieval Conference (TREC-6). National Institutes of Standards and Technology, November 1998, pp. 500-240.

[McCarley 1999]
J.S. McCarley (1999): *Should we translate the queries or the documents in cross-language retrieval?* In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 208-214.

[Chen et al. 1999]
Aitao Chen, Fredric C. Gey, Kazuaki Kishida, Hailing Jiang & Qun Jiang (1999): *Comparing multiple methods for Japanese and Japanese-English text retrieval.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 49-58.

[Chen et al. 2001]
Aitao Chen, Fredric C. Gey & Hailing Jiang (2001): *Berkeley at NTCIR-2: Chinese, Japanese, and English IR experiments.* In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan, March 2001, pp. 137–145.

[Chen et al. 2002]
Kuang-hua Chen, Hsin-Shi Chen, Noriko Kando, Kazuo Kuriyama, Sukhoon Lee, Sung Hyon Myaeng, Kazuaki Kishida, Koji Eguchi & Hyeon Kim (2002): *Overview of CLIR Task at the Third NTCIR Workshop*. In: Working Notes of the Third NTCIR Workshop Meeting, National Institute of Informatics, October 2002, Tokyo, Japan.

[Chen & Gey 2002]
Aitao Chen & Fredric C. Gey (2002): *Experiments on Cross-language and Patent Retrieval at CTCIR03 Workshop*. In: NTCIR Workshop 3: Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Question Answering and Summarization, Tokyo Japan, Oct. 2001–Oct. 2002. NII, Tokyo.

[Chow et al. 2000]
Ken C. W. Chow, Robert W. P. Luk, K. F. Wong & K. L. Kwok (2000): *Hybrid term indexing for different IR models.* In: Proceedings of the fifth international workshop on on Information retrieval with Asian languages. Hong Kong, China, pp 49 – 54.

[Collier et al. 1998]
Nigel Collier, Kenji Ono & Hideki Hirakawa (1998): *An experiment in hybrid dictionary and statistical sentence alignment*. In: Proceedings of the 17th international conference on Computational Linguistics , Vol. 1, Montreal, Quebec, Canada, pp. 268 - 274

[Dumais et al. 1997]
Susan T. Dumais, Todd A. Letsche, Michael L. Littman & Thomas K. Landauer (1997): *Automatic Cross-Language Retrieval Using Latent Semantic Indexing.* In: AAAI Symposium on Cross-Language Text and Speech Retrieval. American Association for Artificial Intelligence, March 1997, pp. 15-21.

[Dumais et al. 1998]
Susan T. Dumais, Thomas K. Landauer & Michael L. Littman (1998): *Automatic cross-linguistic information retrieval using latent semantic indexing*. In: Cross-Language Information Retrieval. Kluwer Academic, 1998.

[Fuji & Croft 1993]
Hideo Fujii, W. Bruce Croft (1993): *A Comparison of Indexing Techniques for Japanese Text Retrieval*. In: Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval Pittsburgh, PA, USA, June 27 - July 1, 1993, pp. 237-246.

[Fujii & Ishikawa 1999]
A. Fujii and T. Ishikawa. *Cross-language information retrieval at ULIS*. In: Proceedings of the 1st NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, pp. 163–169.

[Fujii & Ishikawa 2000]
Atsushi Fujii & Tetsuya Ishikawa (2000): *Applying Machine Translation to Two-Stage Cross-Language Information Retrieval*. In: Proceedings of the 4th Conference of the Association for Machine Translation in the Americas (AMTA-2000), Oct. 2000, pp.13-24.

[Fujii & Ishikawa 2004]
A. Fujii & T. Ishikawa (2004): *Cross-language IR at University of Tsukuba: automatic translateration for Japanese, English, and Korean*. In: Proceedings of the Fourth NTCIR Workshop Meeting, Cross-Lingual Information Retrieval Task.

[Fujita 1999]
Sumio Fujita (1999): *Notes on Phrasal Indexing. JSCB Evaluation Experiments at NTCIR 1 AD HOC.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan.

[Fujita 2001]
Sumio Fujita (2001): *Notes on the Limits of CLIR Effectiveness. NTCIR-2 Evalutaion Experiments at Justsystem.* In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Gey 2004]
Frederic C. Gey (2004): *Chinese and Korean Topic Search of Japanese News Collections*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Gospodnetić & Hatcher 2004]
Otis Gospodnetić & Eric Hatcher (2004): Lucene in Action. Manning, Canada.

[Hackl 2004]
René Hackl (2004): *Mehrsprachiges Information Retrieval im Rahmen von CLEF 2003.* Mag.-Arb., Universität Hildesheim, Informationswissenschaft.

[Hadamitzky 1995]
Wolfgang Hadamitzky (1995): *Handbuch und Lexikon der japanischen Schrift.* Kanji und Kana 1. Langenscheidt: Berlin, München, Wien, Zürich, New York.

[Halpern 2000]
Jack Halpern (2000): *The Challenges of Intelligent Japanese Searching*. Working paper. The CJK Dictionary Institute. Saitama, Japan.

[Halpern 2002]
Jack Halpern (2002): *Lexicon-based Orthographic Disambiguation in CJK Intelligent Information Retrieval*. In: Proceedings of the 19th Conference on Computational Linguistics, COLING-2002, August 24 - September 1, 2002, Taipei, Taiwan.

[Harman 1992]
Donna Harman (1992): *Relevance feedback and other query modification techniques Source*. In: Information Retrieval: Data Structures and Algorithms. Prentice-Hall, Inc.: NJ, USA, pp. 241-263.

[Huang & Robertson 1997]
Hiangji Huang & Stephen E. Robertson (1997): *Experiments on large test collections with probabilistic approaches to Chinese text retrieval*. In: Proceedings of the 2nd International Workshop on Information Retrieval with Aisan Languages 1997 (IRAL'97), October 8-9, 1997, Tsukuba-shi, Japan.

[Jiang & Littmann 2001]
Fan Jiang & Michael L. Littmann (2001): *Approximate Dimension Reduction at NTCIR*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, May 2000- March 2001, Tokyo, Japan.

[Jones et al. 1998]
Gareth J. F. Jones, Tetsuya Sakai, Masahiro Kajiura & Kazuo Sumita (1998): *Experiments in Japanese Text Retrieval and Routing using the NEAT System.* In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, pp 197 – 205.

[Juang & Tseng 2002]
Da-Wei Juang & Yuen-Hsie Tseng (2002): *Uniform Indexing and Retrieval Scheme for Chinese, Japanese, and Korean*. In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, Tokyo, Japan.

[Kanazawa et al. 2001]
Teruhito Kanazawa, Atsuhiro Takasu & Jun Adachi (2001): *R2D2 at NTCIR 2 Ad-hoc Task: Relevance-based Superimposition Model for IR*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Kando et al. 1999]
Noriko Kando, Kazuko Kuriyama, Toshihiko Nozue, Koji Eguchi, Hiroyuki Kato, Souichiro Hidaka (1999): *Overview of IR Tasks at the First NTCIR Workshop.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan.

[Kando 2003]
Noriko Kando (2003): *Overview of the Third NTCIR Workshop.* In: NTCIR Workshop 3: Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Question Answering and Summarization, Tokyo Japan, Oct. 2001–Oct. 2002. NII, Tokyo 2003, pp. 1-16.

[Kang et al. 2004]
In-Su Kang, Seung-Hoon Na & Jong-Hyeok Lee (2004): POSTECH at NTCIR-4: CJKE Monolingual and Korean-related Cross-Language Retrieval Experiments. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Kimura et al. 2004]
F. Kimura; A. Maeda & S. Uemura (2004): *CLIR using Web directory at NTCIR4*. In: Proceedings of the Fourth NTCIR Workshop Meeting, Cross-Lingual Information Retrieval Task, 2004.

[Kishida et al. 2004]
Kazuaki Kishida, Kuang-hua Chen, Sukhoon Lee, Kazuko Kuriyama, Noriko Kando, Hsin-Hsin Chen, Sung Hyon Myaeng & Kochi Eguchi (2004): *Overview of CLIR Task at the Fourth NTCIR Workshop*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Knight & Graehl 1998]
K. Knight & J. Graehl (1998): *Machine Transliteration*. In: Computational Linguistics: 24(4), 1998.

[Kummer et al. 2005]
Nina Kummer; Christa Womser-Hacker; Noriko Kando (2005): *Re-examination of Japanese Indexing: Fusion of Word-, Ngram- and Yomi-Based Indices*. In: Proceedings of the 11th Annual Meeting of The Association for Natural Language Processing, March 14-18, 2005, University of Kagawa, Kagawa Prefecture, Japan, pp. 221-224.

[Kwok & Chan 1998]
K.L. Kwok & M. Chan (1998): *Improving Two-Stage Ad-Hoc Retrieval for Short Queries.* In: Proc 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98). Melbourne, Australia, pp. 250-256.

[Kwok et al. 2004]
Kui-Lam Kwok; Norbert Dinstl & Sora Koi (2004): *NTCIR-4 Chinese, English, Korean Cross-Language Retrieval Experiments using PIRCS*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Levow 2003]
G.-A. Levow (2003): *Issues in pre- and post-translation document expansion: untranslatable cognates and missegmented words*. In: Proceedings of 6th International Workshop on Information Retrieval within Asian languages, pp. 77-83.

[Levow 2004]
G.-A. Levow (2004): *University of Chicago at NTCIR-4: Multi-Scale Query Expansion*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Lin et al. 1999]
Dekang Lin. (1999*): A case-base algorithm for word sense disambiguation*. In: Proceedings of Conference Pacific Association for Computational Linguistics, Waterloo, Canada. Pacific Association for Computational Linguistics.

[Lin et al. 2000]
Du Lin, Zhang Yibo, Sun Le, Sun Yufang & Han Jie (2000): *PM-Based Indexing for Chinese Text Retrieval. In: Proceedings of the fifth international workshop on on Information retrieval with Asian languages*, Hong Kong, China, pp 55 – 59.

[Luk et al. 2001]
Robert W.P. Luk, K.P. Wong & K.L Kwok (2001): *Hybrid Term Indexing: An Evaluation*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan, pp. 130-136.

[Lunde 1999]
Ken Lunde (1999): *CJKV Information Processing*. O'Reilly.

[Mandl & Womser-Hacker 2001]
Thomas Mandl & Christa Womser-Hacker (2001): *Probability Based Clustering for Document and User Properties.* In: T. Ojala (ed.): Infotech Oulo International Workshop on Information Retrieval (IR 2001). Oulo, Finland. 2001, pp. 100-107.

[Matsumura et al. 1999]
Atsushi Matsumura, Atsuhiro Takasu & Jun Adachi (1999): *Structured Index System at NTCIR1: Information Retrieval using Dependency Relationship between Words*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[McNamee 2001]
Paul McNamee (2001): *Experiments in the Retrieval of Unsegmented Japanese text at the NTCIR-2 Workshop*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[McNamee 2002]
Paul McNamee (2002): *Knowledge-light Asian Language Text Retrieval at the NTCIR-3 Workshop*. In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, October 8 – 10, 2002, Tokyo, Japan.

[Mori et al. 2001]
Tatsunori Mori, Tomoharu Kokubu & Takashi Tanaka (2001): *Cross-lingual information retrieval based on LSI with multiple word spaces*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Moulinier 2004]
Isabelle Moulinier (2004): *Thomson Legal and Regulatory at NTCIR-4: Monolingual and pivot-language retrieval experiments*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan, pp. 158-165.

[Murata et al. 2002]
Masaki Murata, Qing Ma & Hitoshi Isahara (2002): *Applying Multiple Characteristics and Techniques to Obtain High Levels of Performance in Information Retrieval*. In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, October 8 – 10, 2002, Tokyo, Japan, pp. 87-92.

[Nakagawa & Kitamura 2004]
Tetsuji Nakagawa & Mihoko Kitamura (2004): *NTCIR-4 CLIR experiments at OKI*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Nakazawa et al. 1999]
Satoshi Nakazawa, Takayoshi Ochiai, Kenji Satoh, and Akitoshi Okumura (1999): *Cross language information retrieval based on comparable corpora*. In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 149-155.

[Nie et al. 1996]
Jian-Yun Nie, Martin Brisebois, Xiaobo Ren (1996): *On Chinese Text Retrieval.* In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, Zurich, Switzerland, pp. 225 – 233.

[Oard & Wang 1999]
Douglas W. Oard & Jianqiang Wang (1999): *NTCIR CLIR Experiments at the University of Maryland*. In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 157-161.

[Ogawa & Iwasaki 1995]
Yasushi Ogawa & Masajirou Iwasaki (1995): *A New Character-based Indexing Organization using Frequency Data for Japanese Documents*. In: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA, July 9-13, 1995, pp. 121-129.

[Ogawa et al. 1993]
Yasushi Ogawa, Ayako Bessho & Masako Hirose (1993): *Simple Word Strings as Compound Keywords: An Indexing and Ranking Method for Japanese Texts*. ACM-SIGIR'93, Pittsburgh, PA, USA.

[Ogawa & Matsuda 1997]
Yasushi Ogawa & Toru Matsuda (1997): *Overlapping statistical word indexing: A new indexing method for Japanese text*. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 27-31, 1997, Philadelphia, PA, USA, pp. 226-234.

[Ogawa & Mano 2001]
Yasushi Ogawa & Hiroko Mano (2001): *RICOH at NTCIR-2*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan, pp. 121-123.

[Ozawa et al. 1999]
Tomohiro Ozawa, Mikio Yamamoto, Kyoji Umemura & Kenneth W. Church (1999): *Japanese Word Segmentation Using Similarity Measure for IR.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 89-96.

[Qu et al. 2003]
Yan Qu, Gregory Grefenstette & David A. Evans (2003): *Automatic Transliteration for Japanese-to-English Text Retrieval.* In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, pp. 353 – 360.

[Qu et al. 2004]
Yan Qu, Gregory Grefenstette, David A. Hull, David A. Evans, Toshiya Ueda, Tatsuo Kato, Daisuke Noda, Motoko Ishikawa, Setsuko Nara & Kousaku Arita (2004): *Justsystem-Clairvoyance CLIR Experiments at NTCIR-4 Workshop.* In: Proceedings of the Fourth NTCIR Workshop on research in Information Retrieval, Automatic Text Summarization and Question Answering, pp.117-122.

[Robertson 1991]
S.E. Robertson (1991): *On term selection for query expansion*. In: Journal of Documentation, 46(4), pp. 59-364.

[Robertson und Sparck Jones 1976]
S.E. Robertson & K. Sparck Jones (1976): *Relevance weighting of search terms*. In: Journal of the American Society for Information Sciences, 27(3). pp. 129-146.

[Savoy 2004]
Jacques Savoy (2004): *Report on CLIR Task for the NTCIR-4 Evaluation Campaign.* In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan, pp. 178-185.

[Sakai 2000]
Tetsuya Sakai (2000): *MT-based Japanese-English Cross-Language IR Experiments using the TREC Test Collections*. In: Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages (IRAL 2000), September 2000, pp. 181-188.

[Sakai et al. 1999]
Tetsuya Sakai, Yasuyo Shibazaki, Masaru Suzuki, Masahiro Kajiura, Toshihiko Manabe & Kazuo Sumita (1999): *Cross-Language Information Retrieval for NTCIR at Toshiba.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 137-144.

[Sakai et al. 2000]
Tetsuya Sakai, Masahiro Kajiura & Kazuo Sumita (2000): *A First Step towards Flexible Local Feedback for Ad hoc Retrieval.* In: Proceedings IRAL 2000, pp. 95-102.

[Sakai et al. 2001]
Tetsuya Sakai, Stephen E. Robertson & Stephen Walker (2001): *Flexible Pseudo-Relevance Feedback for NTCIR-2.* In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan, pp. 59-66.

[Sakai et al. 2002]
Tetsuya Sakai, Makoto Koyama, Masuru Suzuki & Toshihiko Manabe (2002): *Toshiba KIDS at NTICR-3: Japanese and English-Japanese IR*. In: In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, October 8 – 10, 2002, Tokyo, Japan, pp.51-58.

[Sakai et al. 2004]
Tetsuya Sakai, Makoto Koyama & Akira Kumano (2004): *Toshiba BRIDJE at NTCIR-4 CLIR: Monolingual/Bilingual IR and Flexible Feedback*. In: Proceedings of the Fourth NTCIR Workshop on research in Information Retrieval, Automatic Text Summarization and Question Answering, pp.65-72.

[Sato et al. 1999]
Mitsuhiro Sato, Hayashi Ito, Naohiko Noguchi (1999): *NTCIR Experiments at Matshushita: Ad-hoc and CLIR Task.* In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan.

[Sato & Noguchi 2001]
Mitsuhiro Sato and Naohiko Noguchi (2001): *NTCIR-2 Experiments at Matsushita: Monolingual and Cross-Lingual IR Tasks*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Sawada & Umemura 1999]
Ryuichi Sawada and Kyoji Umemura (1999): *Dynamic Programming: A New Paradigm for Information Retrieval*. In: Proceedings of the First NTCIR Workshop

on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan.

[Seo et al. 2004]
H.C. Seo; S.B. Kim; H.G. Lim & H.C. Rim (2004): *KUNLP system for NTCIR-4 Korean-English cross-language information retrieval*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Seo et al. 2004]
Hee-Cheol Seo; Sang-Bum Kim; Ho-Gun Lim & Hae-Chang Rim (2004): *KUNLP System for NTCIR Korean-English Cross-Language Information Retrieval*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Shibatani 1992]
Masayoshi Shibatani (1992): *Japanese*. In: William Bright (Ed.) (1992): *International Encyclopedia of Linguistics*, Vol. 2, Oxford University Press.

[Taylor et al. 1995]
Insup Taylor & M. Martin Taylor (1995): *Writing and Literacy in Chinese, Korean and Japanese (Studies in Written Language and Literacy)*, John Benjamins Publishing Co.

[Takeda et al. 2002]
Yoshiyuki Takeda, Kyoji Umemura, Eiko Yamamoto (2002): *Deciding Indexing Strings with Statistical Analysis.* In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, October 8 – 10, 2002, Tokyo, Japan.

[Tanimura et al. 2001]
Seigo Tanimura, Masashi Suzuki, Hiroshi Nakagawa & Tatsunori Mori (2001): *Japanese and English Cross-lingual Information Retrieval at DLUT.* In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Tomlinson 2004]
Stephen Tomlinson (2004): *Experiments with Decompounded Chinese, Japanese and Korean Words Parsed by Hummingbird SearchServerTM at NTCIR-4.* In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Tsang et al. 1999]
T.F. Tsang, R.W.P. Luk & K.F. Wong (1999): *Hybrid term indexing using words and bigrams*. In: Proceedings of International Workshop in Information Retrieval for Asian Languages (IRAL'99), Taipei, November 11-12 1999, pp112-117.

[Tseng et al. 2004]
Yuen-Hsien Tseng, Da-Wei Juang & Shiu-Han Chen (2004): *Global and Local Term Expansion for Text Retrieval.* In: Proceedings of the Fourth NTCIR Workshop on research in Information Retrieval, Automatic Text Summarization and Question Answering, pp. 73-77.

[Womser-Hacker 1996]
Christa Womser-Hacker (1996): *Das MIMOR-Modell. Mehrfachindexierung zur dynamischen Methoden-Objekt-Relationierung im Information Retrieval*,

Habilitationsschrift, Universität Regensburg.

[Womser-Hacker 2005]
Christa Womser-Hacker (2005): *An Information Retrieval Prototype for Research and Teaching*. To appear in: Maximilian Eibl, Christian Wolff & Christa Womser-Hacker, Christa: Designing Information Systems. Festschrift für Jürgen Krause. Konstanz: Universitätsverlag [Schriften zur Informationswissenschaft].

[Vines & Wilkinson 1999]
Phil Vines & Ross Wilkinson (1999): *Experiments with Japanese Text Retrieval Using mg*. In: Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition, August 30 – September 1, 1999, Tokyo, Japan, pp. 97-100.

[Yang & Ma 2002]
Yiming Yang & and Nianli Ma (2002): CMU in Cross-Language Information Retrieval at NTCIR-3. In: Working Notes of the Third NTCIR Workshop Meeting, Part II: Cross-Lingual Information Retrieval Task, October 8 – 10, 2002, Tokyo, Japan.

[Yoshioka et al. 2001]
Masaharu Yoshioka, Kazuko Kuriyama & Noriko Kando (2001): *Analysis of the Usage of Japanese Segmented Texts in NTCIR Workshop 2*. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization, National Institute of Informatics, Tokyo, Japan.

[Zhang & Vines 2004]
Y. Zhang & P. IVnes (2004): RMIT *Chinese-English CLIR at NTCIR-4*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

[Zhou et al. 2004]
Y. Zhou; J. Qin; M. Chau & H. Chen (2004): *Experiments on Chinese-English cross-language retrieval at NTCIR-4*. In: Working Notes of the Fourth NTCIR Workshop Meeting, June 2-4, 2004, Tokyo, Japan.

List of Links

(all checked on June 17, 2005)

http://trec.nist.gov
http://clef-campaign.org
http://research.nii.acjp/ntcir/workshop
http://www.omniglot.com/writing/japanese_hiragana.htm
http://www.omniglot.com/writing/japanese_katakana.htm
http://en.wikipedia.org/wiki/Japanese_writing_system
http://chasen.aist-nara.ac.jp/hiki/ChaSen/
http://www.jsa.co.jp/EDR/
http://www.csse.monash.edu.au/~jwb-edict.html
http://www.csse.monash.edu.au/~jwb/wwwjdic.html
http://www.csse.monash.edu.au/~jwb/j_jmdict.html
http://www.wadoku.de/
http://bunmei7.hus.osaka-u.ac.jp/download.htm
http://babelfish.altavista.com/
http://yakushite.net/
http://www.google.com/language_tools?hl=en
http://www.freetranslation.com
http://www.tranexp.com:2000/InterTran
http://www.worldlingo.com
http://www.samlight.com/ev
http://www.babylon.com
http://jakarta.apache.org/lucene/docs/index.html
http://chasen.aist-nara.ac.jp/hiki/ChaSen/
http://jgloss.sourceforge.net/

Table of Figures

List of Tables

List of Abbreviations

ADE	Approximate Dimension Equalization
BLIR	Bilingual information retrieval
BRF	Blind Relevance Feedback
CLEF	Cross-Language Evaluation Forum
CLIR	Cross-Language Information Retrieval
CVS	Concurrent Versioning System
DP	Dynamic Programming
EUC	Extended UNIX Code
EUC-JP	Extended UNIX Code for Japanese
FPRF	Flexible Pseudo Relevance Feedback
GDMAX	Maximum Gradient
HMM	Hidden Markov Model
IR	Information Retrieval
JIS	Japan Industrial Standard
KOP	Kanji Overlap Promotion
LSI	Latent Semantic Indexing
MAP	Mean Average Precision
MIMOR	Multiple Indexing for Dynamic Method-Object-Relation in Information Retrieval
MLIR	Multilingual Information Retrieval
MT	Machine Translation
NACSIS	National Academic Center for Science Information Systems
NE	Named Entity
NII	National Institute of Informatics
NLP	Natural Language Processing
NTCIR	NII-NACSIS Test Collection for IR Systems
OOV	Out-Of-Vocabulary
PRF	Pseudo Relevance Feedback
RAM	Random Access Memory
RSV	Retrieval Status Value
	Robertson Selection Value
SGML	Standard Generalized Markup Language
SLIR	Single-Language Information Retrieval
TDF	Text-Document Frequency
TF-IDF	Text Frequency/Inverse Document Frequency
TREC	Text REtrieval Conferences
UTF	Unicode Transformation Format

Appendix A – Stoplists

Table S1: Stopwords for word-based index (Mainichi'98).

Mainichi'98 collection (newspaper articles)		
する	開く	ら
いる	いく	もの
日	見る	さん
れる	たち	これ
なる	日	くる
こと	長	ない
の	発表	できる
人	それ	時
年	おる	円
時間	目	よう
言う	氏	られる
記事	昨年	いう
問題	みる	的
調べる	さ	ため
探す	思う	年
入る	回	人
今年	よる	ある
午後	説明	の

Table S2: Stopwords for word-based index (NTCIR-1).

NTCIR-1 collection (scientific abstracts)		
方式	間	法
物	系	示す
設計	5	研究
持つ	開発	得る
さ	評価	3
0	等	ある
の	46	できる
述べる	必要	検討
効果	考える	1
今回	目的	化
点	明らか	2
有効	解析	ため
本稿	せる	なる
手法	これ	行う
内	よう	性
処理	中	られる
にる	提案	用いる
本	方法	的
高い	可能	結果
問題	場合	いる
時	実験	れる
これら	もの	こと
利用	報告	する
大きい	よる	

Table S3: Stopwords for yomi-based index (Mainichi'98).

Mainichi'98 collection (newspaper articles)		
スル	ショウ	ラレル
イル	カタ	ジュウ
ニチ	オク	テキ
レル	サン	ヨウ
ナル	ジョウ	タメ
コト	イク	イウ
ノ	モンダイ	ニン7
ネン	ヒ	シャ
アル	マエ	カイ
ニン	サ	ネン
サン	キ	アル
イウ	・	ノ
クル	イチ	モノ
ナイ	ヨル	シ
モノ	ラ	?
タチ	ジ	ナナ
イ}	ミル	ハチ
ソウ	クル	ロク
オル	デキル	キュウ
メ	カン	ヨン
ウケル	ナカ	ゴ
ニ	カ	ゼロ
トキ	チュウ	レイ
オモウ	ナイ	サン
ハイ	エン	ニ
テン	チョウ	イチ
ホウ	ケン	

Table S4: Stopwords for yomi-based index (NTCIR-1).

NTCIR-1 collection (scientific abstracts)		
ホンコウ	チ	モト
ケイ	ヒツヨウ	ラ
ゼロ	エイキョウ	ヨル
ド	ヨン	カタ
テン	カンガエル	シメス
リツ	モクテキ	ケンキュウ
ショリ	アキラカ	エル
カンケイ	レイ	ヨウ
コウカ	リョウ	アル
ニル	カイセキ	ケイ
ジカン	コレ	ホウ
スル	カン	サン
カクニン	トキ	デキル
ホン	ガタ	ケントウ
モデル	ネ	モノ
ワカル	ナイ	タメ
タカイ	トウ	カ
モンダイ	セル	ニ
コレラ	チュウ	ナル
リヨウ	ナカ	イチ
ケンキュウ	コウゾウ	ラレル
オオキイ	トクセイ	モチイル
マ	テイアン	セイ
ヒカク	ジュウ	テキ
コウセイ	ホウホウ	ケッカ
タイ	ジョウ	オコナウ
ソクテイ	ゴ	?
モツ	キ	イル
カイハツ	カノウ	レル
ヒョウカ	バアイ	コト
サ	ジッケン	?
ヘンカ	ホウコク	スル
ジ		

Table S5: Stopwords for bi-gram-based index (Mainichi'98).

Mainichi'98 collection (newspaper articles)		
情報	情報	説明
年間	今後	写真
新聞	午前	場合
毎日	社会	時間
一方	以上	今年
可能	時間	日午
現在	今年	真説
今回	昨年	問題
必要	問題	説明
午後		

Table S6: Stopwords for bi-gram-based index (NTCIR-1).

NTCIR-1 collection (scientific abstracts)		
abst	対象	問題
研究	考察	利用
特性	試験	目的
結果	調査	解析
検討	分析	本研
実験	解析	可能
方法	考慮	報告
本稿	本論	提案
方式	論文	場合
設計	重要	必要

Appendix B – Evaluation Results

Basic Indexing Strategies

Table B1: Average precision values of the individual indices per topic (Mainichi '98).

qulD	n-gram	word	yomi
3	.5940	.4254	.4088
4	1.0000	.7715	.7746
5	.3979	.3636	.1886
8	.0860	.0134	.0106
9	.2044	.1580	.1313
10	.6055	.4858	.5156
12	.2720	.4733	.5427
13	.1023	.0686	.1035
14	.4890	.5137	.5045
15	.2232	.0947	.0938
16	.4619	.4396	.4551
17	.0156	.0117	.0092
18	.1326	.0123	.0099
19	.1647	.4275	.3240
20	.3392	.0456	.2937
23	.1373	.1505	.1473
24	.3751	.3878	.3795
26	.2548	.2096	.2342
28	.2571	.1490	.0949
30	.5246	.5717	.5396
32	.0466	.0685	.1146
33	.0064	.0109	.3553
35	.1080	.1834	.1927
36	.5984	.5344	.4814
37	.6305	.4244	.4068
39	.4157	.3862	.4370
40	.6122	.6234	.6392
41	.0807	.1609	.1526
42	.5310	.4513	.4147
43	.5697	.5289	.5132
44	.8537	.8188	.8146
45	.7139	.7174	.7300
46	.7589	.8556	.8356
47	.4680	.4672	.4471
48	.7626	.7717	.7708
49	.3499	.4310	.4602
50	.4787	.4275	.4110
51	.1662	.0308	.0581
52	.2811	.1534	.1274
54	.4621	.6751	.7019
55	.4703	.5119	.5199
56	.2575	.3405	.3195
57	.1065	.2357	.2330
58	.5546	.5346	.5407
59	.4552	.4936	.4809
60	.2045	.1230	.1198
AVG	**.3822**	**.3638**	**.3704**

Table B2: Average precision values of the individual indices per recall level (Mainichi '98).

Recall	N-gram	Word	Yomi
0.0	.7915	.8126	.8287
0.1	.6484	.6650	.6801
0.2	.5818	.5855	.6046
0.3	.5171	.4860	.5045
0.4	.4471	.4245	.4395
0.5	.4020	.3663	.3547
0.6	.3366	.2998	.3050
0.7	.2773	.2370	.2405
0.8	.2118	.1865	.1892
0.9	.1475	.1153	.1144
1.0	.0572	.0249	.0253

Table B3: Average precision values of the individual indices per topic (NTCIR-1).

qulD	n-gram	word	yomi
31	.1708	.2114	.1993
32	.0084	.0500	.0652
33	.5087	.4960	.5051
34	.4478	.1168	.1204
35	.4099	.4066	.4001
36	.1733	.7773	.7984
37	.0719	.0442	.0738
38	.5943	.1019	.0629
39	.3541	.3055	.2770
40	.4198	.3402	.3280
41	.3919	.3720	.3702
42	.2066	.0809	.1267
43	.3713	.2799	.1155
44	.3511	.4248	.4010
45	.0171	.0631	.0790
46	.3377	.2560	.2994
47	.2223	.1657	.2170
48	.1958	.3934	.3564
49	.0201	.0669	.1357
50	.7371	.4807	.6702
51	.0083	.0202	.0133
52	.1020	.2112	.1720
53	.1169	.1030	.0983
54	.0177	.0063	.0044
55	.1668	.1585	.0824
56	.1168	.0834	.2001
57	.5021	.3322	.3529
58	.1729	.0564	.0591
59	.5527	.4809	.4591
60	.6049	.5553	.4579
61	.4746	.3435	.1136
62	.0943	.1298	.1845
63	.1598	.0546	.0718
64	.5769	.6209	.6877
65	.4364	.1963	.2812
66	.5482	.6719	.7224
67	.4387	.5328	.6669
68	.3291	.3450	.3230
69	.0203	.0113	.0138
70	.4423	.4667	.4505
71	.0798	.0506	.0463
72	.1542	.1405	.1435
73	.2341	.3297	.3082
74	.0171	.0147	.0127
75	.0404	.0140	.2119
76	.3283	.1950	.2612
77	.6867	.6407	.7292
78	.3554	.1106	.0611
79	.5765	.1108	.1186
80	.3181	.4619	.5988
81	.0873	.0443	.0618
82	.5846	.5611	.4895
83	.3901	.4104	.3695
AVG	**.2971**	**.2622**	**.2722**

Table B4: 11-point Precision values reached by individual indices per recall level (NTCIR-1).

Recall	N-gram	Word	Yomi
0.0	.7718	.7270	.7569
0.1	.6492	.6098	.6103
0.2	.5319	.4654	.4864
0.3	.4469	.3767	.3731
0.4	.3613	.3263	.3074
0.5	.3018	.2656	.2676
0.6	.2128	.1767	.1944
0.7	.1163	.0932	.1046
0.8	.0730	.0488	.0691
0.9	.0440	.0236	.0318
1.0	.0190	.0074	.0126

Fuzzy Querying

Table FQ1: Avg. prec. basic word-based index vs. Fuzzy Querying (Mainichi'98).

qulD	word	wordf
3	.4254	.4290
4	.7715	.7241
5	.3636	.3636
8	.0134	.0134
9	.1580	.1580
10	.4858	.4858
12	.4733	.4710
13	.0686	.0686
14	.5137	.5212
15	.0947	.0877
16	.4396	.4320
17	.0117	.0117
18	.0123	.0123
19	.4275	.4066
20	.0456	.0308
23	.1505	.1505
24	.3878	.3893
26	.2096	.2096
28	.1490	.1490
30	.5717	.5444
32	.0685	.0670
33	.0109	.0106
35	.1834	.1841
36	.5344	.4285
37	.4244	.4241
39	.3862	.3859
40	.6234	.5441
41	.1609	.1390
42	.4513	.4020
43	.5289	.5040
44	.8188	.7876
45	.7174	.6579
46	.8556	.8556
47	.4672	.4793
48	.7717	.7585
49	.4310	.4297
50	.4275	.4310
51	.0308	.0310
52	.1534	.1534
54	.6751	.8207
55	.5119	.5055
56	.3405	.3402
57	.2357	.2204
58	.5346	.6486
59	.4936	.4727
60	.1230	.1126
AVG	**.3638**	**.3577**

Table FQ2: Avg. prec. basic word-based index and Fuzzy Querying (NTCIR-1).

qulD	word	wordf
31	.2114	.0258
32	.0500	.0106
33	.4960	.4046
34	.1168	.1146
35	.4066	.4467
36	.7773	.6689
37	.0442	.0274
38	.1019	.0352
39	.3055	.0993
40	.3402	.3436
41	.3720	.3611
42	.0809	.0779
43	.2799	.2789
44	.4248	.4326
45	.0631	.0059
46	.2560	.0073
47	.1657	.0190
48	.3934	.2850
49	.0669	.0027
50	.4807	.4662
51	.0202	.0130
52	.2112	.0111
53	.1030	.1001
54	.0063	.0027
55	.1585	.1464
56	.0834	.0608
57	.3322	.3942
58	.0564	.0549
59	.4809	.2430
60	.5553	.5552
61	.3435	.3132
62	.1298	.0970
63	.0546	.0244
64	.6209	.6187
65	.1963	.1055
66	.6719	.6730
67	.5328	.5328
68	.3450	.3491
69	.0113	.0213
70	.4667	.4697
71	.0506	.0512
72	.1405	.1301
73	.3297	.3850
74	.0147	.0147
75	.0140	.0081
76	.1950	.1840
77	.6407	.6314
78	.1106	.1106
79	.1108	.1038
80	.4619	.4668
81	.0443	.0474
82	.5611	.4059
83	.4104	.4085
AVG	**0.2622**	**0.2235**

PRF using Mainichi'98

Table PRF1: Avg. prec. of n-gram-based index with different PRF parameters (Mainichi'98).

qulD	basic n-gram	+D5T30	+D10T30	+D10T20	+D10T40	+D10T50
3	.5940	.6209	.6504	.6507	.6422	.6386
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	.3979	.4156	.4136	.4021	.4130	.4225
8	.0860	.1399	.1409	.1485	.1370	.1342
9	.2044	.1976	.3303	.2625	.3303	.3303
10	.6055	.5786	.6036	.6036	.6036	.6036
12	.2720	.2927	.2630	.2694	.2646	.2623
13	.1023	.0522	.0522	.0522	.0521	.0521
14	.4890	.5131	.5177	.5218	.5223	.5183
15	.2232	.1791	.2142	.2188	.2185	.2258
16	.4619	.4556	.4511	.4466	.4522	.4520
17	.0156	.0761	.0730	.0765	.0722	.0722
18	.1326	.0988	.1039	.1120	.0986	.0962
19	.1647	.1568	.2130	.2194	.2130	.2130
20	.3392	.3204	.3752	.3638	.3780	.3748
23	.1373	.1229	.1320	.1317	.1272	.1297
24	.3751	.3084	.2612	.2799	.2751	.2671
26	.2548	.2551	.2323	.2149	.2294	.2235
28	.2571	.2346	.1861	.1910	.1474	.1466
30	.5246	.5255	.5301	.5170	.5315	.5315
32	.0466	.0895	.0844	.0861	.0845	.0866
33	.0064	.0049	.0049	.0052	.0071	.0058
35	.1080	.1200	.1038	.1014	.1056	.1060
36	.5984	.6286	.5711	.5890	.5753	.5753
37	.6305	.6275	.6255	.6268	.5741	.5729
39	.4157	.4191	.4286	.4224	.4285	.4219
40	.6122	.5908	.6336	.6135	.6075	.6094
41	.0807	.0850	.0812	.0711	.0812	.0765
42	.5310	.7106	.7326	.7673	.7133	.7080
43	.5697	.6404	.6680	.6747	.6508	.6524
44	.8537	.8620	.8620	.8611	.8625	.8632
45	.7139	.8081	.8265	.8190	.8234	.8234
46	.7589	.7263	.6921	.7193	.6881	.6868
47	.4680	.4797	.5075	.4997	.5112	.5083
48	.7626	.8307	.8162	.8122	.8304	.8303
49	.3499	.3896	.3705	.3848	.3637	.3637
50	.4787	.4653	.4613	.4617	.4602	.4602
51	.1662	.1892	.2021	.2074	.1997	.1986
52	.2811	.2673	.2625	.2710	.2620	.2609
54	.4621	.5800	.6323	.6331	.6223	.6165
55	.4703	.4647	.4602	.4606	.4623	.4584
56	.2575	.2557	.2357	.2369	.2381	.2359
57	.1065	.0974	.0954	.0957	.0953	.0975
58	.5546	.6609	.6580	.6538	.6585	.6608
59	.4552	.5124	.5086	.5114	.5004	.5027
60	.2045	.2853	.3125	.2810	.3068	.3017
AVG	**.3822**	**.3986**	**.4039**	**.4032**	**.4005**	**.3995**

Table PRF2: Avg. prec. of word-based index with different PRF parameters (Mainichi'98).

qulD	Basic word	+D5T30	+D10T30	+D10T20	+D10T40	+D10T50
3	.4254	.5161	.4759	.4485	.4523	.4891
4	.7715	.9750	.8582	.8631	.8728	.8648
5	.3636	.4021	.2510	.2589	.2679	.2479
8	.0134	.0106	.0161	.0159	.0148	.0140
9	.1580	.1812	.2095	.1673	.2461	.2508
10	.4858	.4850	.5044	.5092	.5169	.5155
12	.4733	.5123	.5466	.5111	.5418	.5686
13	.0686	.0685	.0516	.0516	.0683	.0683
14	.5137	.5556	.5374	.5413	.5442	.5435
15	.0947	.1696	.1753	.1776	.1732	.1759
16	.4396	.4377	.4322	.4196	.4302	.4315
17	.0117	.0321	.0595	.0674	.0560	.0517
18	.0123	.0141	.0054	.0055	.0101	.0103
19	.4275	.4392	.4948	.4609	.4796	.4985
20	.0456	.0922	.1089	.1124	.1120	.1079
23	.1505	.1686	.1642	.1636	.1731	.1716
24	.3878	.4497	.4278	.4278	.4343	.4429
26	.2096	.2132	.2614	.2120	.2505	.2409
28	.1490	.1882	.0848	.0749	.0885	.0889
30	.5717	.6115	.5845	.5931	.5844	.5900
32	.0685	.1716	.1341	.1268	.1431	.1369
33	.0109	.0152	.0236	.0174	.0234	.0239
35	.1834	.1995	.1928	.1984	.2038	.2076
36	.5344	.5271	.6053	.6108	.5974	.5749
37	.4244	.4374	.3631	.3527	.3605	.3536
39	.3862	.3834	.3901	.3944	.3804	.3709
40	.6234	.6579	.6398	.6281	.6602	.6707
41	.1609	.2201	.2749	.2650	.2518	.2533
42	.4513	.7167	.7415	.7301	.7260	.7148
43	.5289	.6204	.7115	.7065	.7012	.6776
44	.8188	.8248	.8247	.8219	.8242	.8243
45	.7174	.8372	.8158	.8205	.8272	.8204
46	.8556	.9053	.8940	.8905	.8864	.8850
47	.4672	.6155	.6386	.6246	.6295	.6310
48	.7717	.8176	.8423	.8179	.8513	.8518
49	.4310	.4533	.4628	.4597	.4656	.4638
50	.4275	.4175	.4246	.4241	.4240	.4320
51	.0308	.0313	.0300	.0313	.0524	.0721
52	.1534	.1429	.1482	.1438	.1520	.1458
54	.6751	.7630	.8075	.7778	.8038	.8013
55	.5119	.5182	.5098	.5173	.5121	.5000
56	.3405	.3164	.3323	.3211	.3605	.3619
57	.2357	.2476	.2129	.2212	.2078	.2110
58	.5346	.6041	.6338	.6302	.6393	.6395
59	.4936	.5822	.6058	.6041	.6142	.6176
60	.1230	.2045	.1963	.1832	.2230	.2199
AVG	**.3638**	**.4077**	**.4066**	**.4000**	**.4095**	**.4094**

Table PRF3: Avg. prec, of yomi-based index with different PRF parameters (Mainichi'98).

qulD	basic yomi	+D5 T30	+D10 T30	+D10 T20	+D10 T40	+D10 T50	+D10 T60	+D15 T50	+D10 T70	+D10 T100	+D10 T110
3	.4088	.4525	.4710	.4407	.4590	.4712	.4747	.3595	.4644	.5001	.4930
4	.7746	.9519	.7374	.7409	.7748	.7857	.7691	.8471	.7900	.8102	.8070
5	.1886	.1849	.1926	.1967	.1999	.1928	.2030	.2251	.1971	.2102	.2030
8	.0106	.0392	.0811	.0473	.0842	.0623	.0532	.0786	.0464	.0536	.0507
9	.1313	.1403	.1393	.1291	.1554	.1621	.1538	.2275	.1521	.1827	.1833
10	.5156	.5198	.5371	.5371	.5779	.5633	.5676	.5556	.5531	.5523	.5470
12	.5427	.6148	.6201	.6122	.6200	.6114	.6174	.6183	.6255	.6320	.6468
13	.1035	.0697	.0580	.0420	.0476	.0590	.0609	.0361	.0609	.2123	.2124
14	.5045	.5004	.5798	.5584	.5768	.5691	.5679	.6020	.5642	.5525	.5564
15	.0938	.1428	.1412	.1296	.1329	.1306	.1325	.1520	.1271	.1469	.1441
16	.4551	.4122	.4561	.4684	.4416	.4292	.4336	.4765	.4423	.4509	.4609
17	.0092	.0252	.0287	.0226	.0359	.0274	.0246	.0430	.0287	.0333	.0321
18	.0099	.0032	.0032	.0037	.0034	.0031	.0035	.0047	.0031	.0049	.0046
19	.3240	.4099	.4895	.4749	.4782	.4719	.4889	.4669	.4678	.4865	.4733
20	.2937	.3403	.3185	.3316	.3619	.3949	.4142	.3581	.4173	.3817	.3689
23	.1473	.1426	.1242	.1363	.1320	.1293	.1390	.1924	.1400	.1402	.1399
24	.3795	.3776	.4319	.4101	.4737	.4839	.5392	.4867	.5465	.5407	.5614
26	.2342	.2507	.2801	.3019	.2925	.3276	.2939	.2882	.2943	.2634	.2514
28	.0949	.2330	.2158	.1655	.2237	.2462	.2532	.3196	.2238	.2009	.2103
30	.5396	.6277	.6307	.6073	.6421	.6518	.6509	.6269	.6512	.6453	.6477
32	.1146	.2316	.2698	.3109	.2439	.2349	.2173	.2092	.2319	.2060	.1991
33	.3553	.4317	.4284	.4183	.4799	.4716	.4659	.4687	.5019	.5790	.6000
35	.1927	.2766	.3291	.3415	.3099	.3298	.3205	.3543	.3208	.3024	.2791
36	.4814	.4460	.5355	.5325	.5487	.5378	.5220	.5356	.5348	.4679	.4691
37	.4068	.3430	.4110	.4087	.3741	.3696	.3739	.3058	.3785	.3461	.3526
39	.4370	.4571	.4608	.4645	.4510	.4605	.4455	.3736	.4330	.4227	.4334
40	.6392	.7693	.6757	.6126	.7495	.7680	.7896	.6698	.7854	.7940	.7912
41	.1526	.0562	.1128	.1121	.0808	.0802	.0868	.1855	.0873	.0922	.0918
42	.4147	.5703	.6002	.6674	.6276	.6376	.6338	.5475	.6148	.5966	.5887
43	.5132	.5123	.6034	.6308	.6132	.6389	.6488	.6186	.6435	.6042	.5937
44	.8146	.8205	.8126	.8116	.8137	.8140	.8153	.8234	.8176	.8127	.8119
45	.7300	.8610	.8475	.8620	.8541	.8642	.8626	.8552	.8645	.8446	.8411
46	.8356	.9549	.9435	.9284	.9484	.9571	.9588	.9471	.9583	.9438	.9453
47	.4471	.6481	.6632	.6328	.6629	.6628	.6428	.7006	.6358	.6496	.6412
48	.7708	.8278	.8085	.8018	.8189	.8167	.8149	.8267	.8140	.8472	.8401
49	.4602	.5573	.5781	.5257	.5892	.5833	.6055	.5542	.6018	.5790	.5774
50	.4110	.4410	.4602	.4265	.4623	.4638	.4711	.4740	.4722	.4851	.4873
51	.0581	.0509	.0324	.0436	.0285	.0256	.0260	.0283	.0254	.0279	.0329
52	.1274	.0406	.1887	.1548	.1651	.1443	.1336	.2106	.1315	.1194	.1166
54	.7019	.7076	.8695	.8446	.8562	.8480	.8541	.8736	.8860	.8838	.8798
55	.5199	.5333	.5358	.5301	.5431	.5465	.5383	.5624	.5367	.5422	.5407
56	.3195	.4097	.4041	.4091	.3731	.3617	.3643	.3637	.3716	.3683	.3796
57	.2330	.2081	.2272	.2250	.2047	.2114	.2027	.1690	.2068	.2061	.1980
58	.5407	.6912	.6749	.6975	.6760	.6826	.6768	.6607	.6671	.6571	.6607
59	.4809	.6142	.5672	.5538	.5804	.5770	.5804	.6061	.5889	.5990	.6047
60	.1198	.2054	.2389	.2010	.2835	.2844	.2991	.2267	.3004	.2942	.2874
AVG	**.3704**	**.4153**	**.4308**	**.4240**	**.4359**	**.4379**	**.4389**	**.4373**	**.4393**	**.4407**	**.4399**

PRF using NTCIR-1

Table PRF4: Avg. prec. of n-gram-based index with different PRF parameters (NTCIR-1).

quID	n-gram	+D10T30	+D5T30	+D5T50	+D3T30	+D20T30	+D10T10
31	.1708	.0665	.0952	.0952	.0506	.0506	.0665
32	.0084	.0040	.0043	.0043	.0040	.0043	.0040
33	.5087	.6161	.6120	.6120	.5434	.6161	.6161
34	.4478	.5208	.5093	.5093	.5093	.5093	.5208
35	.4099	.4037	.4037	.4037	.4147	.4099	.4037
36	.1733	.2000	.2892	.2892	.2493	.1733	.2000
37	.0719	.0451	.0451	.0451	.0451	.0403	.0451
38	.5943	.6239	.6224	.6224	.6239	.6305	.6239
39	.3541	.3199	.3170	.3170	.3199	.3199	.3199
40	.4198	.4198	.4198	.4198	.4364	.4264	.4198
41	.3919	.3919	.4148	.4148	.3919	.4148	.3919
42	.2066	.2066	.2066	.2066	.2066	.2066	.2066
43	.3713	.5888	.5888	.5888	.4793	.4860	.5888
44	.3511	.3511	.3511	.3511	.3511	.3511	.3511
45	.0171	.0178	.0147	.0147	.0147	.0170	.0178
46	.3377	.3434	.3097	.3097	.3546	.3041	.3434
47	.2223	.2750	.2633	.2633	.2511	.2134	.2750
48	.1958	.1632	.1632	.1632	.1632	.1632	.1632
49	.0201	.0341	.0341	.0341	.0341	.0341	.0341
50	.7371	.6810	.6810	.6810	.6691	.6810	.6810
51	.0083	.0083	.0053	.0053	.0053	.0083	.0083
52	.1020	.1020	.1020	.1020	.1020	.1020	.1020
53	.1169	.1169	.1169	.1169	.1169	.1169	.1169
54	.0177	.0177	.0256	.0256	.0177	.0176	.0177
55	.1668	.1668	.1668	.1668	.1668	.2337	.1668
56	.1168	.1168	.1205	.1205	.1168	.1168	.1168
57	.5021	.4761	.5021	.5021	.4761	.4761	.4761
58	.1729	.1729	.1729	.1729	.1760	.1729	.1729
59	.5527	.5666	.5666	.5666	.5666	.5666	.5666
60	.6049	.6049	.6049	.6049	.6049	.6049	.6049
61	.4746	.4746	.4746	.4746	.4746	.5620	.4746
62	.0943	.0943	.0943	.0943	.0986	.0943	.0943
63	.1598	.1573	.1573	.1573	.1598	.1573	.1573
64	.5769	.5769	.5769	.5769	.5769	.5769	.5769
65	.4364	.4364	.7311	.7311	.4364	.4364	.4364
66	.5482	.5482	.5482	.5482	.5482	.5482	.5482
67	.4387	.4387	.4387	.4387	.4387	.4387	.4387
68	.3291	.3291	.3291	.3291	.3280	.3291	.3291
69	.0203	.0152	.0152	.0152	.0152	.0152	.0152
70	.4423	.4423	.4423	.4423	.4423	.4423	.4423
71	.0798	.0798	.0798	.0798	.0658	.0798	.0798
72	.1542	.1542	.1542	.1542	.1774	.1542	.1542
73	.2341	.2341	.2327	.2327	.2341	.2341	.2341
74	.0171	.0171	.0171	.0171	.0171	.0171	.0171
75	.0404	.0404	.0404	.0404	.0622	.0622	.0404
76	.3283	.3283	.3283	.3283	.3283	.3283	.3283
77	.6867	.6845	.6867	.6867	.6845	.6845	.6845
78	.3554	.3352	.3335	.3335	.3352	.3352	.3352
79	.5765	.6516	.5775	.5775	.6711	.5765	.6516
80	.3181	.3181	.3181	.3181	.3181	.3181	.3181
81	.0873	.0382	.0383	.0383	.0383	.0873	.0382
82	.5846	.5854	.5854	.5854	.5854	.5854	.5854
83	.3901	.3901	.3901	.3901	.3901	.3901	.3901
AVG	**.2971**	**.3017**	**.3079**	**.3079**	**.2998**	**.3004**	**.3017**

Table PRF5: Avg. prec. of word-based index with different PRF parameters (NTCIR-1).

quID	word	+D10T30	+D5T30	+D5T50	+D3T30	+D20T30	+D10T10
31	.2114	.1322	.2114	.2114	.1882	.2854	.1322
32	.0500	.0518	.0512	.0512	.0402	.0433	.0518
33	.4960	.6163	.6080	.6080	.5257	.6163	.6163
34	.1168	.1002	.1168	.1168	.1002	.1002	.1002
35	.4066	.4066	.4066	.4066	.4066	.4528	.4066
36	.7773	.7655	.7584	.7584	.7584	.7655	.7655
37	.0442	.0249	.0249	.0249	.0249	.0249	.0249
38	.1019	.1084	.1084	.1084	.1084	.1084	.1084
39	.3055	.3410	.3170	.3170	.3170	.3191	.3410
40	.3402	.3875	.3854	.3854	.3965	.3402	.3875
41	.3720	.4006	.4006	.4006	.3720	.4006	.4006
42	.0809	.0809	.0809	.0809	.0809	.0809	.0809
43	.2799	.2799	.2799	.2799	.2856	.2799	.2799
44	.4248	.4148	.4237	.4237	.4148	.4148	.4148
45	.0631	.0506	.0506	.0506	.0506	.0575	.0506
46	.2560	.2886	.2948	.2948	.2537	.2640	.2886
47	.1657	.1535	.1535	.1535	.1796	.1535	.1535
48	.3934	.2100	.2100	.2100	.3059	.2100	.2100
49	.0669	.0683	.0683	.0683	.0683	.0683	.0683
50	.4807	.3577	.3566	.3566	.3717	.3747	.3577
51	.0202	.0196	.0067	.0067	.0118	.0236	.0196
52	.2112	.1754	.0921	.0921	.2111	.1754	.1754
53	.1030	.1030	.1315	.1315	.1030	.1315	.1030
54	.0063	.0054	.0054	.0054	.0054	.0054	.0054
55	.1585	.1585	.1585	.1585	.1577	.1585	.1585
56	.0834	.0834	.0834	.0834	.0834	.0834	.0834
57	.3322	.2942	.3322	.3322	.3322	.2942	.2942
58	.0564	.0564	.0564	.0564	.0564	.0564	.0564
59	.4809	.5052	.4809	.4809	.4901	.5052	.5052
60	.5553	.5553	.7079	.7079	.7079	.5553	.5553
61	.3435	.3435	.3435	.3435	.3435	.3435	.3435
62	.1298	.1298	.1298	.1298	.1298	.1298	.1298
63	.0546	.0766	.0766	.0766	.0823	.0546	.0766
64	.6209	.6209	.6209	.6209	.6209	.6209	.6209
65	.1963	.1963	.1414	.1414	.1563	.1963	.1963
66	.6719	.6386	.6384	.6384	.6384	.6243	.6386
67	.5328	.4544	.4544	.4544	.5328	.4544	.4544
68	.3450	.3450	.3440	.3440	.3450	.3721	.3450
69	.0113	.0068	.0068	.0068	.0068	.0068	.0068
70	.4667	.4667	.4667	.4667	.4667	.4667	.4667
71	.0506	.0488	.0488	.0488	.0488	.0506	.0488
72	.1405	.1346	.1346	.1346	.2199	.1346	.1346
73	.3297	.2841	.2841	.2841	.2841	.2135	.2841
74	.0147	.0120	.0120	.0120	.0147	.0120	.0120
75	.0140	.0278	.0278	.0278	.0140	.0278	.0278
76	.1950	.2095	.1959	.1959	.1950	.1950	.2095
77	.6407	.6314	.6407	.6407	.6188	.6314	.6314
78	.1106	.1106	.1106	.1106	.1106	.1106	.1106
79	.1108	.1108	.1108	.1108	.1108	.1108	.1108
80	.4619	.4619	.4619	.4619	.4619	.4619	.4619
81	.0443	.0443	.0497	.0497	.0443	.0443	.0443
82	.5611	.4835	.4835	.4835	.4835	.4835	.4835
83	.4104	.4104	.4104	.4104	.4104	.4104	.4104
AVG	**.2622**	**.2537**	**.2558**	**.2558**	**.2594**	**.2548**	**.2537**

Table PRF6: Avg. prec. of yomi-based index with different PRF parameters (NTCIR-1).

qulD	yomi	+D10T30	+D5T30	+D5T50	+D3T30	+D20T30	+D10T10
31	.1993	.0919	.1317	.1317	.1235	.2706	.0919
32	.0652	.0560	.0586	.0586	.0451	.0697	.0560
33	.5051	.5835	.5803	.5803	.5805	.5835	.5835
34	.1204	.0765	.1121	.1121	.0875	.0645	.0765
35	.4001	.3963	.3963	.3963	.3988	.4298	.3963
36	.7984	.7458	.7355	.7355	.7356	.7458	.7458
37	.0738	.0367	.0367	.0367	.0846	.0367	.0367
38	.0629	.0665	.0728	.0728	.0779	.0665	.0665
39	.2770	.2264	.3042	.3042	.3013	.2679	.2264
40	.3280	.3842	.3748	.3748	.3443	.3262	.3842
41	.3702	.4001	.4001	.4001	.4001	.4001	.4001
42	.1267	.1420	.0822	.0822	.0874	.1420	.1420
43	.1155	.0957	.1052	.1052	.1316	.0692	.0957
44	.4010	.4251	.4360	.4360	.4141	.4360	.4251
45	.0790	.0519	.0595	.0595	.0615	.0638	.0519
46	.2994	.3141	.3101	.3101	.3357	.3283	.3141
47	.2170	.2572	.2158	.2158	.2255	.2737	.2572
48	.3564	.2688	.3558	.3558	.4240	.2688	.2688
49	.1357	.1225	.1220	.1220	.1220	.1225	.1225
50	.6702	.6113	.5890	.5890	.6080	.6310	.6113
51	.0133	.0095	.0091	.0091	.0082	.0095	.0095
52	.1720	.1638	.0966	.0966	.0694	.1638	.1638
53	.0983	.0881	.1252	.1252	.0881	.0881	.0881
54	.0044	.0044	.0052	.0052	.0047	.0044	.0044
55	.0824	.1017	.0993	.0993	.0891	.1017	.1017
56	.2001	.1403	.2295	.2295	.2093	.1753	.1403
57	.3529	.3843	.4250	.4250	.3222	.4050	.3843
58	.0591	.0574	.0574	.0574	.0574	.0574	.0574
59	.4591	.4544	.4250	.4250	.4321	.4516	.4544
60	.4579	.5292	.5292	.5292	.5292	.2921	.5292
61	.1136	.0421	.0351	.0351	.0351	.0521	.0421
62	.1845	.1117	.2055	.2055	.2055	.1125	.1117
63	.0718	.0533	.1151	.1151	.1310	.0551	.0533
64	.6877	.7502	.7502	.7502	.7397	.7421	.7502
65	.2812	.0922	.0922	.0922	.0922	.1163	.0922
66	.7224	.7249	.7242	.7242	.7344	.7361	.7249
67	.6669	.5975	.6434	.6434	.7142	.5878	.5855
68	.3230	.3212	.3190	.3190	.3509	.3290	.3212
69	.0138	.0114	.0091	.0091	.0080	.0126	.0114
70	.4505	.4605	.4611	.4611	.4591	.4565	.4605
71	.0463	.0462	.0462	.0462	.0433	.0471	.0462
72	.1435	.1361	.1832	.1832	.2270	.1372	.1361
73	.3082	.2578	.2578	.2578	.3459	.2578	.2578
74	.0127	.0134	.0144	.0144	.0160	.0060	.0134
75	.2119	.0496	.2487	.2487	.2499	.0496	.0496
76	.2612	.2050	.2340	.2340	.2353	.1882	.2050
77	.7292	.3829	.4162	.4162	.6778	.6197	.3829
78	.0611	.0031	.0043	.0043	.0041	.0008	.0031
79	.1186	.1047	.1059	.1059	.1067	.1047	.1047
80	.5988	.7404	.8385	.8385	.7750	.5620	.7404
81	.0618	.0651	.0861	.0861	.0994	.0548	.0651
82	.4895	.2648	.5571	.5571	.5271	.3178	.2648
83	.3695	.3986	.3986	.3986	.3420	.4027	.3986
AVG	**.2722**	**.2475**	**.2684**	**.2684**	**.2739**	**.2508**	**.2473**

Fusion Runs

Table F1: Avg. prec. of fusion runs with different weights per index (Mainichi'98).

qulD	1N_1W_1Y	2N_1W_3Y	2N_0W_3Y	1N_0W_1Y
3	.5482	.5695	.5679	.5725
4	1.0000	1.0000	1.0000	1.0000
5	.2378	.2217	.2162	.2222
8	.0729	.0783	.0929	.1028
9	.2461	.2235	.2111	.2205
10	.6271	.6342	.6520	.6510
12	.5688	.5672	.5286	.4959
13	.0699	.0711	.1051	.0708
14	.5627	.5591	.5600	.5569
15	.1872	.1811	.1768	.1874
16	.4589	.4615	.4639	.4668
17	.0549	.0475	.0442	.0485
18	.0528	.0457	.0549	.0771
19	.4501	.4732	.4552	.4190
20	.3502	.3680	.3842	.3870
23	.1569	.1481	.1381	.1395
24	.5162	.5298	.5392	.5198
26	.3035	.3246	.3347	.3082
28	.1805	.1902	.1979	.1923
30	.6214	.6302	.6324	.6279
32	.1706	.1805	.1843	.1777
33	.2337	.3967	.4567	.3874
35	.2513	.2691	.2688	.2543
36	.5705	.5556	.5505	.5581
37	.4410	.4394	.4566	.4922
39	.4713	.4816	.4870	.4762
40	.7376	.7623	.7728	.7623
41	.1902	.0941	.0809	.0797
42	.7797	.7524	.7435	.7762
43	.6959	.6781	.6598	.6722
44	.8416	.8394	.8373	.8429
45	.8843	.8858	.8780	.8835
46	.9075	.9202	.9269	.9136
47	.6611	.6662	.6534	.6423
48	.8681	.8698	.8675	.8635
49	.5768	.5892	.5862	.5893
50	.4690	.4752	.4762	.4756
51	.0874	.0705	.0717	.0821
52	.2179	.2226	.2147	.2137
54	.8476	.8689	.8625	.8551
55	.5476	.5552	.5564	.5510
56	.3539	.3604	.3504	.3410
57	.2137	.2030	.1918	.1867
58	.6838	.6816	.6829	.6886
59	.5936	.5924	.5845	.5787
60	.3327	.3286	.3308	.3374
AVG	**.4542**	**.4579**	**.4584**	**.4554**

Table F2: Avg. prec. of fusion runs with different weights per index (NTCIR-1).

qulD	1N_1W_1Y	3N_1W_1Y	2N_1W_1	3N_1W_2Y	1N_0W_1Y
31	.2071	.1963	.1986	.1952	.1940
32	.0463	.0339	.0394	.0399	.0375
33	.5188	.5261	.5240	.5236	.5253
34	.3022	.3953	.3727	.3722	.3640
35	.4271	.4234	.4244	.4245	.4210
36	.6978	.5023	.6283	.6096	.5664
37	.0762	.0768	.0786	.0795	.0818
38	.3944	.5675	.5305	.5154	.4768
39	.3770	.3519	.3950	.3946	.3804
40	.3860	.4081	.4025	.4048	.4049
41	.4101	.4091	.4118	.4102	.4085
42	.1196	.1584	.1470	.1504	.1607
43	.3024	.3638	.3365	.3182	.2599
44	.4422	.4451	.4477	.4475	.4395
45	.0516	.0307	.0374	.0394	.0426
46	.3748	.3947	.3887	.3960	.3912
47	.2625	.2360	.2363	.2375	.2396
48	.2920	.2286	.2452	.2353	.2246
49	.0566	.0304	.0397	.0421	.0520
50	.6936	.7112	.7046	.7088	.7183
51	.0122	.0085	.0098	.0097	.0091
52	.1405	.1456	.1530	.1442	.1517
53	.1285	.1467	.1436	.1378	.1341
54	.0097	.0098	.0104	.0103	.0095
55	.1543	.1663	.1651	.1612	.1485
56	.1432	.1304	.1355	.1402	.1536
57	.5359	.5236	.5284	.5337	.5294
58	.0704	.0846	.0766	.0769	.0746
59	.5129	.5399	.5301	.5271	.5202
60	.6085	.6066	.6070	.6073	.6073
61	.4900	.5076	.5053	.4990	.4540
62	.1100	.1024	.1041	.1041	.1033
63	.1534	.1630	.1605	.1601	.1508
64	.6131	.5883	.5989	.6050	.6068
65	.3564	.4325	.3584	.3709	.3839
66	.5684	.5283	.5550	.5604	.5678
67	.5518	.5089	.5324	.5421	.5500
68	.3323	.3300	.3310	.3273	.3281
69	.0150	.0181	.0167	.0171	.0178
70	.4628	.4571	.4596	.4561	.4541
71	.0584	.0682	.0634	.0625	.0598
72	.1420	.1500	.1466	.1466	.1476
73	.3257	.3135	.3293	.3159	.2701
74	.0162	.0161	.0168	.0164	.0160
75	.0778	.0633	.0729	.0830	.1547
76	.2739	.3053	.2927	.2860	.2843
77	.6973	.7008	.7008	.7034	.7292
78	.3250	.3538	.3534	.3534	.3532
79	.5314	.5772	.5574	.5323	.5321
80	.4982	.3937	.4140	.4155	.3996
81	.1121	.1058	.1018	.0999	.1041
82	.5824	.5984	.6079	.6046	.5816
83	.4203	.4078	.4193	.4175	.4107
AVG	**.3107**	**.3121**	**.3141**	**.3127**	**.3092**

Acknowledgments

ありがとう

… meinen Eltern, dass ich immer tun darf, was ich will und dabei immer mit Unterstützung rechnen kann.

… Prof. Dr. Christa Womser-Hacker für die Hilfe bei der Organisation und Vorbereitung und für die Unterstützung während der Arbeit.

… Prof. Dr. Noriko Kando for inviting me to the NII, for her kind support, for enriching discussions, and for a great time in Tokyo and Takamatsu.

… dem DAAD für die finanzielle Unterstützung.

… Axel für die ewigen Diskussionen, den kritischen Blick und die vielen Ideen.

… to the people from 14th floor for a great working environment and a lot of nice coffee breaks (including some people from 13th floor).

… to the Badminton Team for distraction and exercise.

… Lucie für die große Hilfe im Stress zwischendurch.

… Daniel, der mir ein organisiertes Vorbild war.

… Britta für Frühstücks- und andere Pausen und Stephen für die aufmerksame Korrektur.

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Hildesheim, im Juni 2005

_____